BIOMARKER DISCOVERY AND CLINICAL OUTCOME PREDICTION USING KNOWLEDGE-BASED BIOINFORMATICS

A Dissertation
Presented to
The Academic Faculty

by

John H. Phan

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
Wallace H. Coulter Department of Biomedical Engineering

Georgia Institute of Technology
Emory University
May 2009

BIOMARKER DISCOVERY AND CLINICAL OUTCOME PREDICTION USING KNOWLEDGE-BASED BIOINFORMATICS

Approved by:

Dr. May D. Wang, Advisor
Department of Biomedical Engineering
Georgia Institute of Technology and Emory University

Dr. Andrew N. Young
Pathology and Laboratory Medicine
Emory University School of Medicine

Dr. Xiaoping Hu
Department of Biomedical Engineering
Georgia Institute of Technology and Emory University

Dr. Brani Vidakovic
Department of Biomedical Engineering
Georgia Institute of Technology and Emory University

Dr. Russell M. Mersereau
School of Electrical and Computer Engineering
Georgia Institute of Technology

Date Approved: 3 March 2009

ACKNOWLEDGEMENTS

I would like to first thank my adviser, Dr. May D. Wang, for supporting me over the last several years and allowing me to pursue my strengths as a researcher. At the same time, I am also grateful that she pushed me to pursue collaborative projects with the FDA and caBIG that she correctly predicted to be high-impact and that greatly improved the quality of my work.

I would also like to thank the members of the Bio-MIBLab–graduate students, post-doc, and undergraduate students–with whom I've closely collaborated on several projects. I appreciate the contributions of several graduate and undergraduate students that temporarily worked in the lab, helping me with various projects ranging from software development to conference publications.

My committee members–Xiaoping Hu, Russell Mersereau, Brani Vidakovic, and Andy Young–have given me helpful feedback and have taken time out of their busy schedules to examine and critique my research progress. I am grateful for this guidance.

Finally, I would like to thank my parents, sisters, and brother for their love and support and for not asking me *too* many times when I will be graduating. ☺

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

Advances in high-throughput genomic and proteomic technology have led to a growing interest in cancer biomarkers. These biomarkers can potentially improve the accuracy of cancer subtype prediction and subsequently, the success of therapy. However, identification of statistically and biologically relevant biomarkers from high-throughput data can be unreliable due to the nature of the data-e.g., high technical variability, small sample size, and high dimension size. Due to the lack of available training samples, data-driven machine learning methods are often insufficient without the support of knowledge-based algorithms. We research and investigate the benefits of using knowledge-based algorithms to solve clinical prediction problems. Because we are interested in identifying biomarkers that are also feasible in clinical prediction models, we focus on two analytical components: feature selection and predictive model selection. In addition to data variance, we must also consider the variance of analytical methods. There are many existing feature selection algorithms, each of which may produce different results. Moreover, it is not trivial to identify model parameters that maximize the sensitivity and specificity of clinical prediction. Thus, we introduce a method that uses independently validated biological knowledge to reduce the space of relevant feature selection algorithms and to improve the reliability of clinical predictors.

Biologically relevant feature selection algorithms are those that favor independently validated biomarkers. We show that guiding feature ranking algorithm and parameter selection using these biomarkers improves the efficiency of detecting new

biomarkers that are also likely to validate. Furthermore, the algorithm selection process iteratively evolves as it learns and incorporates new biomarkers into the knowledge set. Using both maximum likelihood and maximum *a posteriori* approaches, we show that the choice of an optimal or biologically relevant method changes in the presence of knowledge feedback. The clinical utility of biomarkers depends on their feasibility in clinical prediction applications. Thus, in a similar approach as–and in collaboration with–the FDA Microarray Quality Control (MAQC) Consortium, we examine several microarray datasets to assess the effect of knowledge-guided feature selection on prediction accuracy. The microarray datasets in our study vary in sample size and clinical focus. For each clinical focus–renal cancer, prostate cancer, and breast cancer–we build and test classification models using independent training and testing datasets in order to reduce prediction bias. Results of these experiments indicate that knowledge-guided feature selection improves clinical prediction. Finally, one of the primary obstacles in translating research to clinical applications is the inaccessibility of bioinformatics applications to the general community of clinicians and biologists. Therefore, we implement several functions of the knowledge-based framework as a web-based and user-friendly application called omniBiomarker. We develop functions of omniBiomarker according to standards of the NCI Cancer BioInformatics Grid (caBIG), further increasing the overall impact of this work.

CHAPTER I

INTRODUCTION

Traditional medical techniques are subjective, especially for clinical problems such as cancer subtype classification. This subjectivity often limits the effectiveness of therapy. For example, cancers with similar morphologic characteristics may behave very differently under similar treatment conditions [50]. Furthermore, some cancer subtypes are more likely to recur after completion of treatment [119, 96]. Thus, recent diagnostic research focusing on these issues involves identification of diagnostic markers–called biomarkers–that have the potential to increase accuracy and remove subjectivity from cancer diagnosis. This relatively new approach to medical diagnostics has become possible with the advent of high-throughput technology for measuring biomolecular expression–e.g., genes, proteins, and other biomolecules. Essentially, this technology has allowed us to take snapshots of the functional state of a biological process. From these snapshots, which typically contain large quantities of data, we can identify the components that are most correlated with clinical symptoms, and then assemble the components into a cohesive story that may explain the internal biological mechanisms of a disease. Most importantly, we can use some of these components as clinical predictors of disease state.

However, the analytical procedure for high-throughput data is not without pitfalls. We call this analytical procedure the biomarker identification pipeline. It consists of several steps with multiple possible solutions at each step: 1) data acquisition, 2) quality control and normalization, 3) feature selection, 4) validation and biological interpretation, and 5) clinical prediction (**Figure 1**). From a purely data-driven perspective, the primary concern for this pipeline is that it can be highly sensitive to the

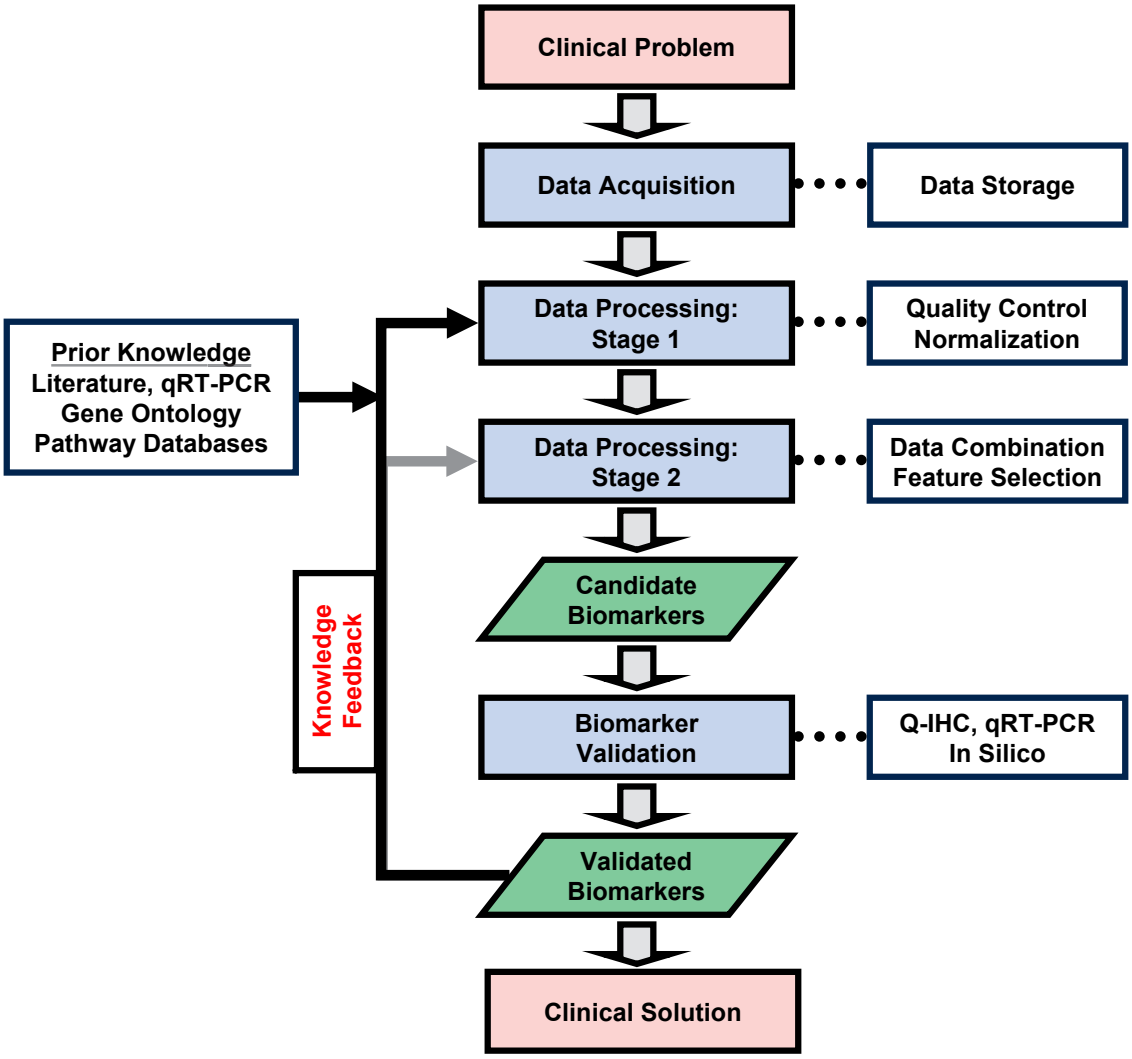


Figure 1: Biomarker Identification Pipeline. The translational bioinformatics pipeline includes several steps that guide us from a clinical problem to a clinical solution. The first step involves acquisition of relevant data, normally in the form of large quantities of genetic or proteomic expression data and associated patient history. Before analyzing the data to identify differentially expressed biomarkers, we must assess and improve the quality of the data by removing technical artifacts and combining multiple datasets to improve statistical relevance. Stage two of data processing involves knowledge-guided identification of relevant, differentially expressed biomarkers. We obtain the knowledge from external sources as well as through iterative feedback to improve the efficiency of biomarker identification. Finally, before using biomarkers in the feedback process or in a clinical application, we must validate their functional relevance by directly assessing expression with alternate assay technologies such as IHC or qRT-PCR.

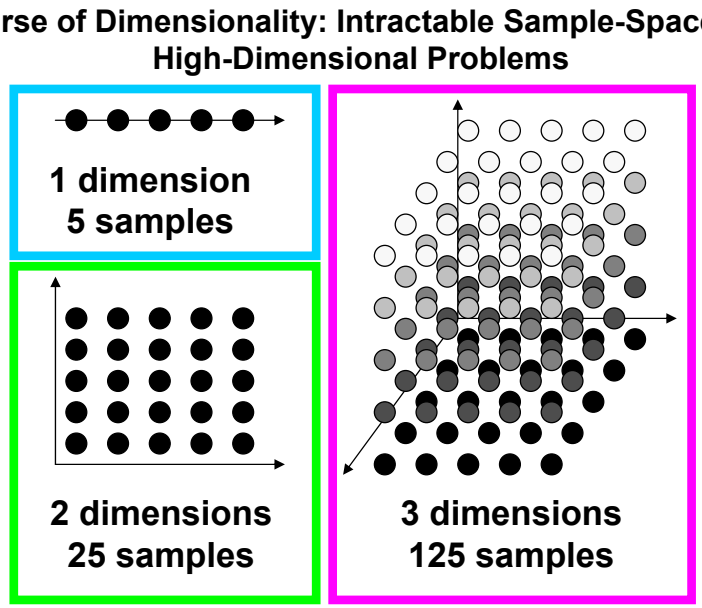


Figure 2: Curse of Dimensionality. The sample space within a unit interval increases exponentially as dimension increases. This is problematic for learning algorithms because a larger number of samples is required to adequately cover the sample space and accurately represent the population.

analysis parameters due to the nature of the data (small sample size relative to high dimension, known as the "curse of dimensionality" problem) [134, 92]. Furthermore, because of the multiple solutions at each step, there are many possible paths in the pipeline, each of which can produce a different result. As such, it is generally not reproducible. We cannot expect a single path to perform well for all clinical scenarios [41]. Rather, we should use specific analysis parameters for each clinical problem. Aside from exhaustively searching every possible analytical path, how do we determine the best parameters for a particular clinical scenario? We address this problem by describing methods to integrate information and increase the amount of available data for solving clinical problems. In addition to combining data from heterogeneous sources, we introduce a method that uses prior biological knowledge to choose an analytical path in the biomarker identification pipeline. Specifically, we focus on the following steps in the pipeline: feature selection, biomarker validation, and estimation of prediction accuracy.

In order to better convey the difficulty of high-throughput data analysis, we describe the "curse of dimensionality" problem. This problem is most apparent in the prediction component of the pipeline, which is a classic pattern recognition problem. Typically, we train a prediction rule from a finite number of input samples in order to classify future samples–in this case, microarray samples, which are the most common high-throughput data platform. The more thoroughly the input samples cover the space of all possible samples–i.e., the more samples available–the more accurately the prediction rule will be able to classify future samples drawn from the same population. Clinical prediction from microarray samples is problematic for two previously mentioned reasons: small sample size and large sample dimension. Each microarray sample contains thousands of genes and clinical microarray samples are difficult to obtain in sufficiently large quantities to adequately represent the population. Although the availability and cost of microarray samples has improved, we must consider this growth with respect to the dimension size of each sample. The size of the input space increases exponentially when the number of dimensions in each sample increases. For example, consider a one-dimensional interval of unit length (**Figure 2**). We can sample at least five points from this space such that the points are evenly spaced at a distance of no greater than 0.2 from adjacent points. If we increase the dimension of this space to two–a square–the minimum number of evenly spaced samples required to cover the space increases to 25. At 10 dimensions, this number increases to 5^{10}. Thus, as the dimension size increases, we need more samples to adequately represent the population. Microarrays are an extreme case. With a limited sample of the population, resulting clinical prediction rules will likely over-fit to a small fraction of the population. Consequently, these prediction rules will be unable to generalize to the population as a whole, resulting in poor prediction performance. Bioinformaticians handle this problem by reducing the dimensionality of the data, usually by removing or selecting features based on their potential predictive ability. However, the curse of

dimensionality also affects feature selection from high-throughput data.

Every step in the biomarker identification pipeline affects the final result: accuracy of the clinical prediction rule. However, assuming that the acquired data is of reasonable quality, the feature selection step is arguably the most important. The features not only affect the complexity of the resulting classifier, but they also provide us with some clues about the underlying biology of the clinical problem. Therefore, it is not surprising that feature selection is one of the more difficult steps of the pipeline. Feature selection algorithms are optimization problems that search for the best set of genes–in the case of microarray data–with the highest potential for accurate prediction. Because gene expression is not independent, feature selection algorithms must identify groups of genes that act in concert [150]. Here, we must again tackle the curse of dimensionality problem, this time in the context of optimization. The space of possible gene combinations increases exponentially as the number of dimensions increases. For example, consider the problem of searching for a single gene from a set of N total genes. Here, we need only evaluate N genes to find the best predictor. But suppose we are interested in a pair of genes. In this case, we need to evaluate $N!/[2!(N-2)!]$ total gene pairs. In the case of choosing K genes out of N total, the number of unique groups grows exponentially to $N!/[K!(N-K)!]$. K is also unknown, although some studies have tried to determine the optimal number of genes for maximizing prediction accuracy [57]. Indeed, there is no existing algorithm that can find the true optimal set of genes from a typical gene expression dataset. The problem is simply too computationally intractable. As such, there are many algorithms that approximate the optimal solution from a reduced search space [31]. However, these approximate gene lists are highly variable and depend on both the algorithm and the data. Fortunately, we may still be able to solve this problem by using knowledge-based methods.

Consider the problem of guessing a shape given only four points randomly sampled

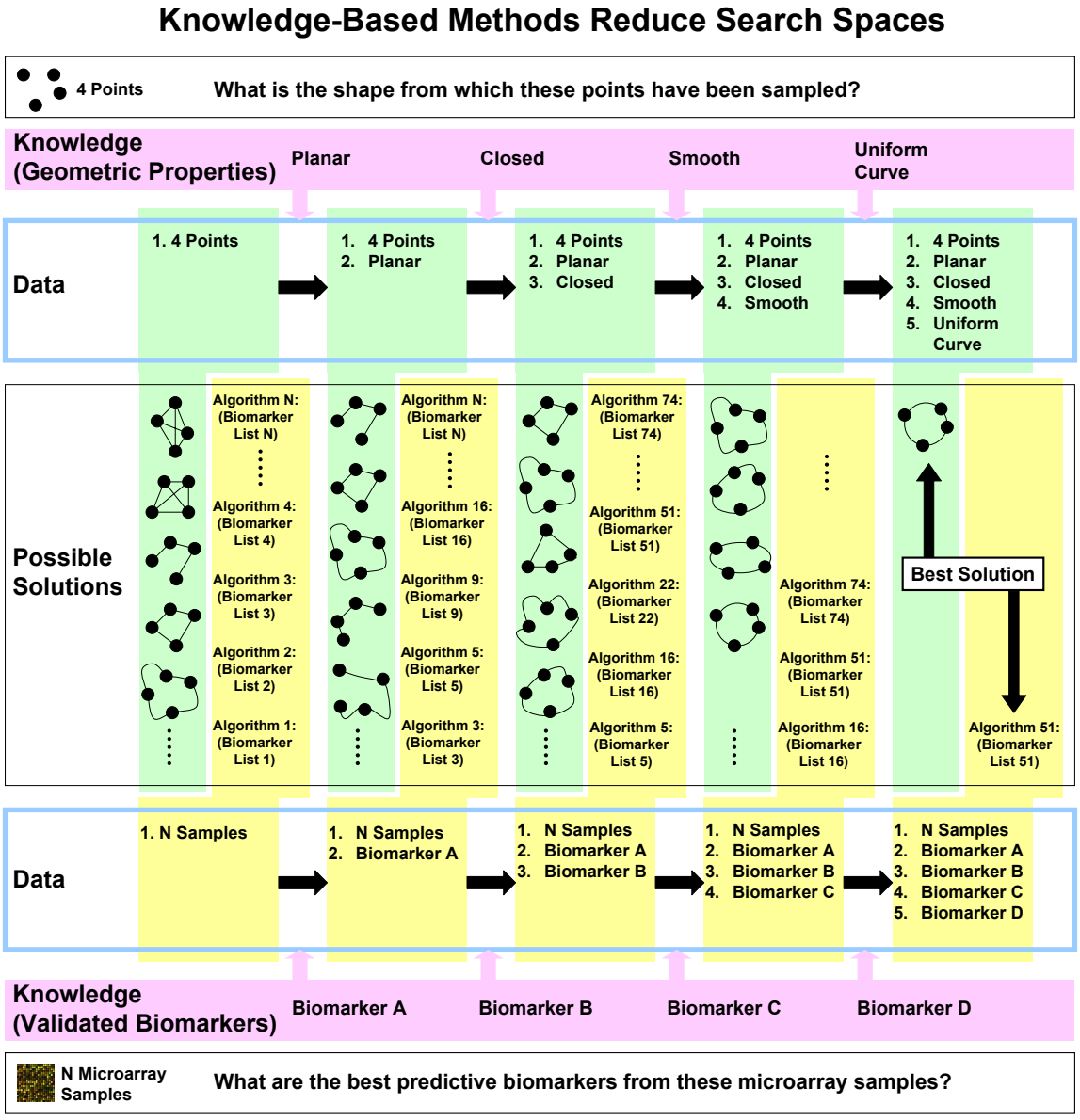


Figure 3: Knowledge-Based Methods. A simple example of search space reduction is guessing a shape given only a few data points (top). As information about the geometric shape is introduced, the number possible solutions shrinks. Eventually the true shape emerges as the only solution. Likewise, the biomarker search space is large due to the variety of available feature selection algorithms, each of which produces a different list of candidate biomarkers (bottom). As information about true biomarkers is introduced, the number of valid feature selection algorithms reduces. The reduced population of algorithms includes those that can consistently identify the known biomarkers.

from the surface of the shape (**Figure 3**). This is impossible because there are infinite possibilities. However, if we can obtain additional clues about the shape, we can begin to narrow the possibilities. For example, if we know that these four points exist on a single plane, then we can exclude three dimensional shapes. If we are told that the shape is closed and smoothly curved, we can reduce the possibilities even further. Finally, if we know that the curvature of the shape is uniform, then there is only one possibility: a circle. Similarly, given a set of microarray samples drawn randomly from a population of patients, there are an exponential number of potential gene lists, varying due to the many algorithms as well as patient heterogeneity. However, if we know that relevant feature selection algorithms should favor specific biomarkers, then we can narrow the list of algorithms and subsequently, the number of potential solutions.

The integration of knowledge in the biomarker identification pipeline is essential for overcoming the curse of dimensionality, stabilizing the variability of feature selection and prediction results, and increasing the overall reproducibility of the process. As indicated in (**Figure 1**), we can obtain prior knowledge from external sources or through a feedback mechanism after the validation step. Each of the following chapters represents work that has been published or is in preparation for publication in peer reviewed journals or conference proceedings. In **Chapter 2**, we introduce previous work in the areas of microarray technology and existing analytical methods. Particularly, we review several studies that have relied on high-throughput gene expression technology to better understand biological mechanisms of disease. Because high-throughput technology is not without hazards, we review several studies that examine these hazards, primarily pertaining to statistical quality of the data and resulting analytical challenges. Portions of this chapter have been summarized in an article that reviews knowledge-based bioinformatics and applications of these methods to biomarker identification and cancer nanotechnology [105]. As we have

noted, one of the primary problems in high-throughput data analysis is the lack of samples. **Chapter 3** introduces a method to combine heterogeneous microarray data in order to improve the statistical significance of subsequent analyses [104]. **Chapter 4** introduces a method to improve the biological relevance of feature selection from high-throughput data and, at the same time, increase reproducibility of the analytical results [106]. **Chapter 5** is a large scale application of these ideas into a framework of clinical prediction. Here, we examine the feasibility of microarrays for clinical prediction. We focus primarily on the notion of biological relevance: does the biological relevance of analytical methods such as feature selection correlate to higher predictive performance? The work in **Chapter 5** was inspired by our collaboration with the FDA which produced several manuscripts currently under review [127, 102, 68]. Finally, in **Chapter 6**, we introduce a translational bioinformatics software application that aims to deliver knowledge integration technology to a wider audience of clinicians and biologists. This software application, called omniBiomarker, is currently under review for caBIG silver level compatibility [95].

CHAPTER II

PREVIOUS WORK

The primary clinical motivation for using high-throughput data is disease therapeutics. Many studies have used microarray technology to identify differentially expressed disease biomarkers using clinically derived patient data. These biomarkers serve as diagnostic and prognostic indicators that subsequently determine therapeutic choices. This chapter reviews the literature focusing primarily on applications of microarray technology to cancer. Specifically, we identify studies that search for differentially expressed genes with the aim of cancer prediction or clinical subtype discovery. The clinical community uses a very diverse set of analytical techniques. As such, we also review techniques that correlate to steps in the biomarker identification pipeline: quality assessment, feature selection, data combination, and prediction.

For the reasons mentioned in the previous chapter, bioinformatics techniques have increasingly gravitated to knowledge-driven techniques. Thus, as an introduction to our own techniques, we also review recent developments in knowledge-based bioinformatics. Finally, because the use of bioinformatics in cancer applications is an interdisciplinary endeavor, we review the many attempts to deliver bioinformatics algorithms to the broader community of clinicians and biologists through software applications.

2.1 Identifying Cancer Biomarkers Using Microarray Technology

The effectiveness of cancer treatment depends not only on early detection, but also on accurate determination of the best therapeutic regimen. Pathologists examine morphologic characteristics of cancer specimens to determine the subtype or stage of the

disease. Physicians then select an appropriate course of treatment based on this information. The goal is to treat a patient's cancer with minimally invasive procedures while maximizing the patient's comfort and chances for survival. Early detection markers for some cancers have been successful while others are still under investigation. For example, prostate specific antigen (PSA) screening has improved the early detection of prostate cancer [133]. Some studies have also identified markers able to detect ovarian and breast cancer at early stages [130, 129, 49]. However, tailoring treatment to individual patients has been much more difficult. As a result, treatment is not always minimal and some patients relapse. For example, chemotherapy and hormonal treatment can prevent breast cancer metastasis, but these treatments are not always necessary [143, 16]. Furthermore, of all men who elect surgery after detecting prostate cancer at an early stage, many are low risk and did not require the surgery while almost one third relapse [133]. Some cancers, such as those of the kidney, are very heterogeneous and include several histopathological subtypes with distinct clinical and cytogenetic features [34]. Consequently, malignant renal cancers are often difficult to treat, as clinical behavior is hard to predict [80]. These difficulties in tailoring treatment to individual patients are due to the limitations in classifying cancers based on morphologic characteristics and to the lack of accurate biomarkers with high correlation to clinical outcomes.

Cancers with similar morphologic characteristics may behave very differently despite similar treatment conditions [50]. In order to improve the prediction of clinical outcomes, medical scientists have searched for other aspects of cancer not readily apparent during morphologic inspection. Naturally, because cancer is the result of genetic mutations, therapeutic research has primarily focused on molecular differences between clinically relevant subtypes. Engineers have developed technology for quickly quantifying molecular expression and biologists have used this technology (e.g. microarrays and mass spectrometry) to generate large quantities of clinical data

[133, 143, 19, 124, 67, 81, 144, 116, 148]. Although these technologies produce data faster than the rate at which we can thoroughly analyze and interpret them, biologists have still identified many candidate biomarkers. Singh *et al.* obtained a large number of prostate cancer microarray samples consisting of several histologic and clinical subtypes. They were able to identify markers correlated with Gleason score (a measure of the cancer's stage, or level of severity) and markers that predict patient outcome following prostatectomy (removal of the prostate). However, the number of samples with available clinical outcome data was relatively small and the authors did not test the predictive performance of these markers using independent data [133]. Chandran *et al.* used similar microarray samples to identify differential markers for distinguishing two histologic classes of prostate cancer from normal tissue. Their study examined the heterogeneity of prostate cancer and the effects of cancers on adjacent normal tissue [19]. Varambally *et al.* used both proteomic and genomic assays to identify markers for metastatic prostate cancer. They found protein markers with high correlation to mRNA expression for predicting clinical outcome [144]. van't Veer *et al.* identified a panel of gene expression signatures that predicted metastasis after diagnosis of breast cancer [143]. Liu *et al.* identified a 186-gene invasiveness gene signature (IGS) that predicted metastasis and survival in breast cancer patients. They showed that this gene signature was applicable to other types of cancer as well [81]. Schuetz *et al.* examined microarray gene expression data for several renal tumor subtypes. They identified many genes up-regulated for specific tumor subtypes compared to all other samples. They verified some markers using qRT-PCR (quantitative Reverse Transcription Polymerase Chain Reaction, an RNA amplification technique used to verify gene expression [142]) and IHC (immunohistochemistry, a method for quantifying and localizing proteins in a tissue sample using fluorescent antibodies [27]) [124]. Jones *et al.* identified biomarkers for similar renal tumor subtypes in addition to a metastatic subtype using a larger microarray dataset [67]. In both renal tumor studies, markers

for distinguishing chromophobe (CHR) and oncocytoma (ONC) subtypes were difficult to identify because of histologic similarity between samples and small sample size. Rohan *et al.* specifically addressed this issue and identified biomarkers to distinguish CHR and ONC renal tumor subtypes [116]. High-throughput genomic and proteomic technology have improved biomarker identification for predicting cancer subtype and clinical outcome. However, concerns regarding technology and protocol standardization, data quality, data analysis methods prevent widespread use of this technology in clinical settings [45].

2.2 Assessing the Quality of Microarray Data

Although many of the previously cited works have successfully used microarrays, this technology is still subject to many technical issues. These issues include microarray quality, sample size, and feature size. Despite a major concern for the quality of high-throughput microarray technology, recent studies have shown that microarray technology is reproducible across laboratories and platforms [126]. For example, Canales *et al.* compared five microarray platforms to three quantitative, low-throughput, gene expression technologies, including TaqMan, standardized qRT-PCR, and QuantiGene assays [17]. They found that all platforms generally agree with quantitative results with the exception of a few genes due to weak expression or differences in probe sequence. Furthermore, Patterson *et al.* compared one- and two-color microarray platforms in terms of reproducibility, sensitivity, specificity, and accuracy and found that these platforms are identical [103]. Two-color microarray data needs special consideration when assessing data quality. Several software packages exist for two-color microarray data preprocessing, including arrayMagic [14]. On the other hand, statistical artifacts in microarray data sometimes appear as spatially correlated regions on arrays with abnormal hybridization. Specifically for Affymetrix arrays, some algorithms, such as RMA Express, have been designed to compute gene

expression and to visualize spatial artifacts [7, 62, 79, 135]. caCORRECT identifies spatial artifacts with image processing algorithms and reduces their effect on gene expression estimation using a modified quantile normalization method [137]. caCORRECT also assigns quality scores to microarray chips, allowing us to remove samples that are below a specified quality threshold. Similarly, the quality of microarray hybridization experiments can be assessed as deviations of gene expression distributions from uniformity [12]. Several methods exist for normalization of microarray data. These methods greatly affect downstream differential expression analysis and should be carefully applied [54, 56, 63, 125]. Some of these normalization methods address limitations in microarray design that introduce unwanted data correlations [70]. Despite the general reproducibility of microarrays and the available software for detecting and correcting statistical artifacts, the detection of differentially expressed biomarkers is still difficult when data sample size is small and feature size is high. Furthermore, statistical artifacts tend to be enhanced when sample size is small, resulting in biased biomarker detection. Bioinformaticians have designed many feature selection algorithms to specifically handle these problems.

2.3 Feature Selection and Gene Ranking Methods

Feature selection algorithms generally fall into two categories: filter and wrapper methods [151, 61]. Filter methods, e.g., fold-change, signal to noise ratio, and variants of the t-test, rank genes by computing a score representing the degree of differential expression. Many studies have explored both parametric and non-parametric filtering methods. Fold-change was the method of choice in many early biological studies involving microarrays. Later research used parametric statistical tests such as the t-test, which estimates Gaussian parameters for each class and computes the significance of differential expression as a measure of error probability. However, the statistical soundness of such simple methods is questionable, as these methods require

strict and unrealistic assumptions about the data distributions. The problem with the t-test, specifically, is the assumption of normality, which is not always true for gene expression data [140]. In situations where data are not normally distributed, non-parametric methods are more appropriate and can improve the results of biomarker detection. For example, Troyanskaya *et al.* compared the standard t-test to a non-parametric t-test that computes p-values using several thousand data permutations [140]. They also examined the non-parametric Wilcoxon rank sum test, which ranks biomolecule expression values and compares the mean rank between classes. Significance analysis of microarrays (SAM), developed by Tusher *et al.*, is a modified t-test that also uses permutations to estimate p-values and selects biomolecules after adjusting for the false discovery rate (FDR) [141]. Despite the success of filtering methods, these methods are still lacking when applied to problems involving multi-dimensional or non-linearly expressed genes. To address this, Lu *et al.* used Hotelling's T^2 test, a generalized multi-dimensional t-test, and identified differentially expressed gene combinations [84]. Biomarkers identified with filter methods are not necessarily accurate classifiers [151]. Thus, much focus has turned to developing feature selection methods that are also accurate estimators of classification performance.

Wrapper methods combine feature selection and classification by ranking genes based on estimated classification accuracy [61, 151]. Because the primary application of biomarker identification is to classify future samples, we place emphasis on wrapper methods. Ranking genes based on classification accuracy better ensures the performance of identified biomarkers in diagnostic applications. Wrapper methods usually involve training a classifier with a finite set of labeled samples and testing the classifier by predicting the class labels of an independent set of samples. The results of testing are normally used as a ranking metric. In contrast to filter methods, wrapper methods are easily applied to multi-dimensional and non-linear data because of the variety of available classifiers.

The major obstacles for wrapper methods are model selection and sample size. By carefully addressing these issues, we can avoid incorrectly estimating classification errors for feature selection. Incorrect estimates can lead to false identification or ignorance of relevant biomarkers [10]. Model selection involves choosing the appropriate classifier and parameters in order to maximize the classifier's ability to generalize. The number of features, or dimension size of samples, also affects the ability of a classifier to generalize. Usually, there is an optimal number of features for specific classifier and data combinations [57]. Parameters that affect classifier complexity, e.g. the radial basis kernel parameter, should be fine-tuned to individual datasets. As the complexity of a classifier increases, training error tends to decrease. However, this decrease in training error is also accompanied by an increase in testing error [52]. We can identify classifier parameters corresponding to optimal testing error by using cross validation methods.

Several studies have examined the effect of small sample size on wrapper methods. This problem, inherent in high-throughput expression data, can cause high variability in estimated classification errors. Subsequently, this leads to high variability in ranking results among different methods [10, 128]. Braga-Neto *et al.* compared cross validation, resubstitution (training error), and bootstrap methods using simulated small sample expression data [10]. They found that cross validation is highly variable due to variations in small sample data resulting from hold-out testing. Although the high variability problem does not exist for resubstitution–resubstitution is low-biased, i.e. classification error estimates are lower than the true error. Bootstrap estimators have a lower variance and are unbiased, but are computationally expensive [10, 44]. Braga-Neto *et al.* proposed a new method, called bolstering, to correct estimation bias due to small sample size while maintaining computational efficiency [9]. Additionally, Fu *et al.* showed that bootstrap cross validation (BCV) performs better than both bootstrap and cross validation, but is very computationally intense [44]. Each

of these methods may be appropriate under certain data conditions and should be selectively considered depending on available computational resources.

2.4 Combining Heterogeneous Microarray Datasets

In order to alleviate the sample size problem, we often want to combine high-throughput expression data from separate, but similar studies. For example, we can combine Affymetrix microarray data at a low level using software applications such as RMA Express [7], dChip [79], or caCORRECT [137] provided that all microarray samples are of the same platform. Some of these software applications include quality control in the gene expression computation process. However, data can be difficult to combine because of technological variability in cases where there is no standard protocol.

Because there is no standard microarray platform, different studies often use different platforms resulting in gene expression measurements that are not directly comparable [24]. Even experiments using the same platform may produce different expression values due to variations in assay protocols or in samples between laboratories. Several studies have attempted to combine microarray data after computing gene expression values. For example, Park *et al.* used ANOVA to identify sources of variation between microarray experiments from different laboratories that use the same platform [100]. Once they adjusted their model to account for unwanted variations, they identified differentially expressed biomarkers in datasets from three different laboratories. Choi *et al.* combined gene expression values from multiple microarray studies by computing the effect size of variations comparable between datasets [24]. The benefits of combining microarray datasets have been illustrated using cancer gene expression data. Xu *et al.* identified differentially expressed biomarkers for prostate cancer using several datasets and Wang *et al.* did the same for leukemia [152, 147]. Phan *et al.* extended the method by Wang *et al.* to wrapper-based methods using a bootstrap approach to combine ranking estimates weighted by the standard deviation

of the measurement [104].

When we are limited to only a small amount of data, we may only identify some of the relevant biomarkers. Furthermore, different datasets may produce different lists of relevant biomarkers. By combining several smaller datasets, we not only increase the statistical significance of biomarker selection, but also gain a more global perspective of the problem.

2.5 Predictive Diagnostic and Prognostic Models

There are several studies that examine the general feasibility of disease classification using microarrays [111, 88, 97, 131, 33, 77]. Some of these studies focus on the analytical and machine learning aspects while others focus on the clinical utility. Dudoit *et al.* used three microarray datasets to test a variety of classifiers including Fisher linear discriminant, maximum likelihood discriminant, nearest neighbors (KNN), classification trees, and aggregating classifiers [33]. Their results indicate that simple classifiers such as KNN and linear discriminants generally perform as well or better than more complicated classifiers. In fact, many microarray and related high-throughput data classification studies have used the simple KNN classifier [118, 23, 75, 109]. Ntzani *et al.* reviewed 84 studies that attempted to correlate clinical cancer outcomes to gene expression profiling. Results from these 84 studies were variable. However, because microarray studies were relatively new at the time, data sample sizes were small and validation of results rare [97]. The reviews by Simon and Quackenbush *et al.* examine key steps and common pitfalls involved in building clinically predictive models [111, 131]. Notably, Simon stresses the importance of correctly estimating the accuracy of prediction models on future samples. This involves proper division of samples into training and testing sets prior to any analysis such that resulting prediction models will not have "seen" any information about testing samples. The

study by Michiels *et al.* reinforces this recommendation after they re-analyzed several large cancer prediction studies [88]. Their results indicated that many of these studies predicted no better than random chance. They discovered that the selection of features greatly depends on the samples and recommended a method of repeated random sampling to better estimate the mean and variance of prediction error. Generally, the correct protocol of building a predictive model is well established and the pitfalls of the process repeatedly outlined [132, 2, 145]. However, even following the protocols may lead to poorly performing models due to the vast set of parameters and algorithms available at each step of the process.

2.6 Improving Biological Relevance by Integrating Knowledge

Some investigators have attempted to improve feature selection by using biological knowledge. Purely data-driven feature selection methods require a sufficient number of patient samples in order to adequately cover the problem space. Because the problem space for high-throughput data is exponentially large, feature selection algorithms that rely on both data and knowledge tend to perform better [149]. There are some examples of this phenomenon in the literature that are specifically targeted at feature selection [106, 21, 107]. Many other examples exist that are somewhat related.

Aerts *et al.* combined data from several resources to prioritize genes relevant to diseases of interest [1]. Their data sources included Gene Ontology databases, published literature, microarray repositories, and sequence information. They extracted "training" genes–genes tagged as differentially expressed in or related to the biological problem–from these databases and ranked test genes according to their similarity to the training genes. Kuffner *et al.* identified groups of genes that simultaneously correlate to genes mentioned in relevant literature and to differential components of

expression profiles [72]. Kong *et al.* searched for combinations of genes that are differentially regulated based on multivariate Hotelling's T^2 statistic and that correlate with Gene Ontology and other pathway databases [71]. Mukherjee *et al.* developed a theoretical framework to compare feature ranking metrics in the presence of control features [93]. Chen *et al.* modified an independent component analysis (ICA) method for detecting biomarkers using inferred biological knowledge [21]. They showed that their knowledge-guided method improved the efficiency of detecting biomarkers compared to traditional ICA methods. Both of these studies are similar to the study by Phan *et al.*, in which biological knowledge in the form of validated biomarkers were used to identify the best feature ranking method out of a population of methods. The authors showed that a knowledge-guided iterative approach to feature selection improved the efficiency of identifying relevant biomarkers [106].

2.7 Bioinformatics Systems and Software Applications

Biomarker identification generally follows a commonly established sequence of steps– the biomarker identification pipeline. In order to understand the scope of this pipeline and the obstacles that prevent its use in clinical practice, we summarize the pipeline with examples of existing solutions for each step. Primarily, we focus on bioinformatics tools available as web-based applications. These tools are developed with a focus not only on new algorithms, but also on the integration of multiple analytical methods into a user-friendly, web-accessible interface. Indeed, many new bioinformatics applications generally do not introduce new algorithms. Instead, they focus on the usability and accessibility of their application, an attribute that is increasingly important as the gap between clinical applications and bioinformatics narrows.

The first step in the biomarker identification pipeline is quality control. Due to the stochastic nature of high-throughput data, it is important to assess data quality prior to analysis. Moreover, the large quantity of high-throughput data requires specialized

software applications. There are several existing applications that assess data quality, particularly for microarrays, within a population of samples while simultaneously estimating and normalizing gene expression. These applications vary in terms of modeling complexity and usability, ranging from downloadable software packages– RMA Express [62], dChip [78]–to web-based applications such as caCORRECT [137]. Although gene expression assays are generally reproducible [126], statistical artifacts in smaller datasets should be identified and either corrected or removed prior to further data analysis.

Early in the microarray era, bioinformatics tools often focused on unsupervised clustering, reflecting researchers' interest in exploring a new technology and discovering interesting properties in the structure of the data without dwelling on potential clinical applications. For example, Eisen *et al.* developed a software application that combines several types of unsupervised clustering methods [38]. AMIC@, a more recent development, also focuses exclusively on clustering. However, AMIC@ combines clustering algorithms as well as visualization tools into a web-based application [48]. Clustering algorithms have not evolved significantly to this day. However, we have increasingly applied clustering to high-throughput gene expression data from many different clinical scenarios. These investigations have led to significant findings in clinical applications, especially in studies concerning cancer subtype identification [30, 29]. As such, unsupervised clustering applications are still widely used for data visualization and discovery.

More recently, the focus of microarray analysis has shifted away from unsupervised clustering to the more guided and powerful supervised analysis. Consequently, web-based bioinformatics applications have also shifted. These new tools focus on genes differentially expressed between known conditions. Some are specific to microarray platforms. For example, MAGMA and ILOOP are web-based applications designed to analyze two-channel microarrays [114, 108]. ILOOP is an interface that assists

in experimental design of two-channel microarrays while MAGMA incorporates standard normalization and statistical methods into an application whose primary aim is usability and reproducibility. Not surprisingly, many of these web-based applications implement functionality for several common steps in the data analysis pipeline. GEPAS (Gene Expression Profile Analysis Suite), for example, includes functions that address several aspects of microarray analysis, including data normalization, feature selection, class prediction, and even unsupervised clustering [139]. CARMAweb is yet another recent development for microarray analysis [113]. CARMAweb uses modules from Bioconductor, an open-source bioinformatics software package that leverages the R programming language. The microarray analysis functions in Bioconductor include background correction, quality control, normalization, differential gene detection, clustering, dimensionality reduction, and visualization [47]. Again, as with most bioinformatics applications, CARMAweb's contribution to the bioinformatics community is an integration of many tools into a user-friendly web interface. Yet another compilation of many gene expression analysis tools is GenePattern [115]. GenePattern, however, furthers the concept of usability and reproducibility by integrating with the cancer Bioinformatics Grid (caBIG), an initiative by the National Cancer Institute (NCI) to create a standard for semantic interoperability of bioinformatics software [95]. Despite the existence of many web-based tools for biomarker identification, we are still several steps removed from clinical applications. Before using these clinical biomarkers in clinical scenarios, we must interpret and verify their biological validity.

It is known that lists of candidate biomarkers from feature selection studies depend on both the available samples as well as the selection algorithm [88]. In fact, these lists may be highly unstable. Furthermore, high-throughput assay platforms typically consist of tens of thousands of genes, many of which are still not fully understood. Thus, the task of interpreting these results is daunting. By associating

each candidate gene to a biological function, we can begin to understand 1) the underlying mechanisms of the associated disease and 2) the biological relevance of the feature selection algorithm. Databases such as the Gene Ontology (GO) are designed to facilitate interpretation on a large scale [46]. However, there is no single method to extract statistically significant conclusions from a GO database analysis. Analogous to quality control, clustering, and feature selection algorithms, GO tools are diverse. Some of these tools, all available as web-based or downloadable software, including GoMiner [154, 155], GOStat [4], AmiGO [18], BiNGO [86], and GOEAST [157]. Each of these tools varies in statistical methodology for determining over-representation of functional GO categories as well as in usability and available software features. Fortunately, the community has recognized the difficulty in choosing a particular software package, resulting in applications such as SerbGO [91]. SerbGO assists researchers by narrowing the list of existing GO applications depending on their specific data analysis needs. There are also similar applications that mine literature rather than the GO database. CoPub, for example, links lists of candidate genes to keywords obtained from literature in Medline abstracts and visualizes statistically over-represented keywords using a network structure [43].

With the steady accumulation of large amounts of gene expression data, several applications have emerged that organize and integrate these data sources and heterogeneous datasets more effectively. As previously mentioned, increasing data sample size is a way to increase the reproducibility of the resulting predictive models. Thus, there has been a demand for data sharing solutions. The Gene Expression Omnibus (GEO) [35] and ArrayExpress [101] are examples of large repositories that adhere to community data standards such as MIAME [11]. ArrayWiki is an alternative solution that allows the user community to annotate gene expression metadata [138]. caArray is part of the caBIG initiative and is intended to become a semantically interoperable

standard for microarray storage for caBIG applications [95]. Just as there is an overlap of analytical methods in gene expression analysis software and gene interpretation software, there is an overlap of data in these high-throughput data repositories. Consequently, we are not surprised to acknowledge the existence of an application called the 'Microarray Retriever' [66]. This web-based application retrieves gene expression data from both the GEO and ArrayExpress repositories in order to maximize the potential for large-sample microarray studies. Similarly, GEOmetadb is an improvement on the querying capabilities of the GEO repository [158]. Although it is currently only available for GEO, it is easy to see that meta-analysis applications are becoming increasingly useful.

The availability of many software packages for each step in the biomarker identification pipeline enables us to choose from a variety of methods to suit our needs. However, the lack of an established data standard impedes our progress when we try to fit the pieces together [98]. For example, without translating the data format, we may not be able to use the data output of a quality control and normalization application in a subsequent clustering or feature selection application. Furthermore, we sometimes need to translate lists of gene symbols from a feature selection application before interpretation with a particular GO application. Workflow applications such as GeneTrailExpress and Taverna address this issue in different ways. GeneTrailExpress is a comprehensive web-based application that implements its own normalization, statistical analysis, interpretation, and visualization modules based on common methods [69]. Taverna is more general and builds workflows for caBIG certified web services [59]. The goals of these workflow applications support those of translational research by speeding up the process by which bioinformaticians can assess the clinical feasibility of a particular data-specific workflow. Furthermore, these workflows may potentially simplify the analytical process for clinicians by establishing predefined algorithm parameters known to work well for particular clinical situations.

CHAPTER III

IMPROVING MICROARRAY DATA SAMPLE SIZE BY COMBINING DATASETS

3.1 Introduction

Microarray gene expression profiling has become a popular tool for identifying biomarkers in diseases such as cancer. They allow us to examine the expression levels of thousands of genes in a single experiment. However, cancer may only alter a tiny portion of the human genome. Therefore, we need a powerful and effective feature selection scheme, in addition to a large sample size, to identify these potential biomarkers. While the number of gene expression datasets available to the scientific community is growing, the sample size of each dataset remains small compared to the number of features. As such, methods for combining multiple datasets have the potential for increasing the power of microarray data analysis by pooling information.

Combining datasets can be difficult when we use different microarray platforms or apply different probe normalization and summarization techniques. Even when we use the same microarray hardware and software, the laboratory effect can, in some cases, be more significant than the choice of microarray platform when assessing reproducibility [64]. Differences in reproducibility, sensitivity, and specificity between datasets from separate test sites can lead to different sets of candidate biomarkers [103, 146]. In addition to all of these technical obstacles, the practical limitation of finding datasets which measure the same scientific question further hampers data combination. Thus, most current biomarker identification studies are limited to single, small-sample datasets.

A common goal in microarray analysis is the creation of predictive classifiers. The

first step in creating a classifier is often feature selection, which involves systematically excluding a number of weakly-informative genes in order to increase the overall performance of the classifier. Methods for feature selection fall into two categories: filter methods and wrapper methods.

Filter methods are a two step process, beginning with individual scoring of each feature, followed by selection based on this scoring. At the end of the filtering procedure, we build a predictive classifier using a different method from the one used to score and select individual genes. Common filtering methods include fold change and t-test. However, the classification accuracy of biomarkers resulting from such methods is not necessarily high. Because of the inclusion of redundant information, resulting classifiers may become highly complex without significant gain in accuracy [151]. Furthermore, these methods are sensitive to small-sample data and depend on strict assumptions. Calculation of the t-statistic, for example, breaks down when the number of features included is larger than the sample size. Statistics such as mean and variance may be significantly biased when calculated from small sample data, leading to false conclusions of significance. The dependence of the t-test on data normality is also problematic, since this assumption is often not true for gene expression data [140].

For wrapper methods, the final classifier is intrinsic to the feature selection process. Instead of scoring genes independently, a wrapper method will assess groups of genes based on their synergistic performance, usually measured by estimating the error-rate of classification. Using classification error-rate as a selection criterion is appropriate when the aim is to design a discriminant rule [150]. Furthermore, error estimation techniques such as the bootstrap do not depend on assumptions of data normality. Studies have shown that various bootstrap and cross validation resampling methods are accurate estimators of predictive performance for small-sample data [10].

Several studies examine methods for combining multiple microarray datasets in

order to improve sample size. These methods include large-scale data-mining, functional integration, ANOVA models, and effect size meta-analysis [100, 147, 60, 153]. Of particular interest is the study by Wang *et al.*, which combined the fold change of genes between classes from three microarray datasets. Their computed statistic also accounted for the different variances within datasets [147]. Choi *et al.* computed a combined z-statistic and found that their combined statistic identifies more potential biomarkers than those of single datasets [24]. None of these methods, however, explored data combination using wrapper-based gene selection methods that estimate classification error. Such methods would have both the benefits of accurate identification of predictive genes from small datasets as well as increased sample size by combining multiple datasets.

The meta-analytic method that we propose combines heterogeneous microarray datasets by combining bootstrap estimated classification errors for each gene. Our method, adapted from Wang *et al.*, weights the combined classification error from each dataset by the inverse of its bootstrap variance [147]. This weighting reduces the overall contribution of datasets with large variance. It is easily extended to any number of datasets and has the potential to improve the biological and statistical relevance of candidate biomarkers.

3.2 Methods

3.2.1 Microarray Data

We use two groups of microarray datasets to test the bootstrap meta-analysis method. The first group includes samples from two renal cancer studies. Each study contains samples from clear cell (CC), chromophobe (CHR), and oncocytoma (ONC) renal cancer subtypes. We are interested in identifying genes that are differentially expressed between the CC and the combined group of both CHR and ONC subtypes. The smaller dataset from Schuetz *et al.* contains 13 CC, 4 CHR, and 3 ONC samples.

The larger dataset from Jones *et al.* contains 32 CC, 6 CHR, and 12 ONC samples. Samples from both datasets are derived from Affymetrix microarrays and contain at least 8793 genes [124, 67]. The Jones dataset contains more genes, but we reduce the number of genes to a subset of 8793 common to both datasets.

The second group includes two prostate cancer datasets with 12625 genes on Affymetrix microarrays. Between the two datasets, there are a total of 113 tumor samples and 113 normal samples [133, 19]. Normal samples are extracted from tissue adjacent to prostate tumors. The smaller Singh dataset contains 52 tumor samples and 50 normal adjacent samples. The Chandran dataset contains 61 tumor samples and 63 normal adjacent samples.

3.2.2 Feature Ranking

We use a wrapper-based approach to rank genes by classification accuracy, estimated using a linear support vector machine (SVM) classifier and bootstrap resampling [28]. The SVM classifier predicts the class of a test sample based on its gene expression by constructing a maximal margin discriminating hyperplane from training samples. We use a dataset-specific SVM cost parameter, C, of approximately $1/n$. Thus, the cost parameter is 0.05 for the Schuetz dataset and 0.02, 0.01, and 0.01 for the Jones, Singh, and Chandran datasets, respectively.

The true classification error for a gene, k, given a classifier is $E_{\cdot,k}$. However, the true error is unknown since we are limited to a finite sample space within a dataset, j. We can compute an estimate of the true classification error $E_{j,k}$ for a dataset, j, and gene, k, using a bootstrapping method. For a dataset with $n = 1 \ldots N$ samples, the bootstrap algorithm randomly selects N samples with replacement [36]. The unique N^* samples selected by bootstrapping ($N^* < N$) are designated as the training set, $S_{training}$, while the remaining $N - N^*$ samples are the testing set, $S_{testing}$.

We define the m^{th} estimate, $E_{j,k,m}$, of the true error of classification computed

with training and testing sets randomly partitioned into $S^m_{training}$ and $S^m_{testing}$. m enumerates all possible partitions of the data into training and testing sets. To estimate $E_{j,k}$, we repeatedly permute the training and testing sets and compute the average of all classification errors:

$$E_{j,k,m} = \frac{\sum_{n \in S^m_{testing}} |L^m_{j,n} - L_{j,n}|}{N - N^*} \tag{1}$$

Here, $L^m_{j,n}$ is the predicted class label of sample n from the testing set $S^m_{testing}$ after training with the set $S^m_{training}$. $L_{j,n}$ is the true class label of the sample n in dataset j. We consider only binary classification, in which the labels may take the values of 0 or 1. We estimate the true classification error using $B = 100$ bootstrap permutations:

$$E_{j,k} = \frac{1}{B} \sum_{m=1}^{B} E_{j,k,m}. \tag{2}$$

Likewise, the variance of the bootstrap estimate is

$$\sigma^2_{j,k} = \frac{1}{B} \sum_{m=1}^{B} (E_{j,k,m} - E_{j,k})^2. \tag{3}$$

In wrapper-based feature selection, we can use the mean error estimates for feature ranking and variance estimates as measures of confidence in these classification errors.

3.2.3 Data Combination

Differences in sample size prevent us from directly comparing wrapper-based feature selection between two or more datasets. The distribution of over all genes k often depends on the balance of samples between classes within a dataset as well as the classifier parameters. Furthermore, the total number of samples in a dataset affects $\sigma^2_{j,k}$. Thus, we need to empirically estimate a null distribution of classification errors for genes in each dataset in order to generate a universally comparable (and combinable) significance score. A common method for estimating the null distribution, $E^0_{j,k}$, is to randomly permute the dataset class labels for each gene and re-compute

the bootstrap, removing any information that the true class labels hold. Normally, we should compute multiple permutations for each gene to estimate a gene-specific null distribution. However, because we are interested in the null distribution across all genes, one permutation per gene is sufficient and computationally efficient. The null classification error for a single bootstrap iteration is

$$E^0_{j,k,m} = \frac{\sum_{n \in S^m_{testing}} |L^{m*}_{j,n} - L^*_{j,n}|}{N - N^*} \tag{4}$$

where $L^{m*}_{j,n}$ and $L^*_{j,n}$ are randomly permuted class labels. The corresponding mean and variance over $B = 100$ bootstrap iterations are $E^0_{j,k}$ and $s^0_{j,k}$. Over all k genes, the sample mean and variance of the null distribution are $\bar{E}^0_j$ and $\hat{s}^0_j$. We can now normalize any bootstrap estimated error $E_{j,k}$ against its null distribution such that its distribution is approximately normal with zero mean and unit variance:

$$Z_{j,k} = \frac{1}{\hat{s}^0_j}(E_{j,k} - \bar{E}^0_j) \tag{5}$$

The variance of a bootstrap measurement $\sigma^2_{j,k}$ estimates the reliability of that measurement. Generally, as sample size increases, bootstrap variance decreases. Thus, when combining bootstrap measurements from multiple data sources for a single gene, we should favor measurements with low variance. Therefore, the combined classification score is a weighted combination of normalized bootstrap errors from individual datasets. We adapt this combination formula from Wang *et al.* [147]:

$$E_{comb,k} = \frac{\sum_j Z_{j,k}/\sigma^2_{j,k}}{\sum_j 1/\sigma^2_{j,k}} \tag{6}$$

with a corresponding null distribution composed of the elements:

$$E^0_{comb,k} = \frac{\sum_j Z^0_{j,k}/s^2_{j,k}}{\sum_j 1/s^2_{j,k}} \tag{7}$$

where $Z^0_{j,k}$ is the normalized null distribution:

$$Z^0_{j,k} = \frac{1}{\hat{s}^0_j}(E^0_{j,k} - \bar{E}^0_j). \tag{8}$$

We can identify differentially expressed genes from individual datasets by ranking values of $Z_{j,k}$ for all genes k or by estimating p-values using the estimated null distribution $Z^0_{j,k}$. Similarly, we can identify significant genes from combined datasets by ranking values of $E_{comb,k}$ for all genes k, or by estimating p-values using the estimated null distribution. The empirical p-value for a gene k from an individual dataset j is

$$p_{j,k} = \frac{1}{\ell}\sum_{i=1}^{\ell} I(Z^0_{j,i} < Z_{j,k}) \tag{9}$$

where ℓ is the total number of genes in the dataset. Likewise, the empirical p-value for a gene k from several combined datasets is

$$p_{comb,k} = \frac{1}{\ell}\sum_{i=1}^{\ell} I(E^0_{comb,i} < E_{comb,k}). \tag{10}$$

In addition to the bootstrap error method described above, we combine data sources on the basis of fold change and t-test as described by Wang *et al.* [147]. **Figure 4** summarizes the data combination scheme.

3.2.4 Validating the Relevance of Gene Ranking

We assess the effect of data combination by comparing the result of feature selection in each individual dataset to that of the combined dataset. It is often difficult to compare feature selection between methods or similar datasets due to the large number of features and the subjectivity of feature interpretation. Because of our limited understanding of biological mechanisms and the noise inherent in microarray technology, we can often only verify the validity of a selected gene by independent assays such as qRT-PCR [25]. However, in order to avoid the costly and time-consuming validation of genes selected in this study, we use a method introduced by Mukherjee *et al.* to compute the probability of successfully selecting differentially expressed

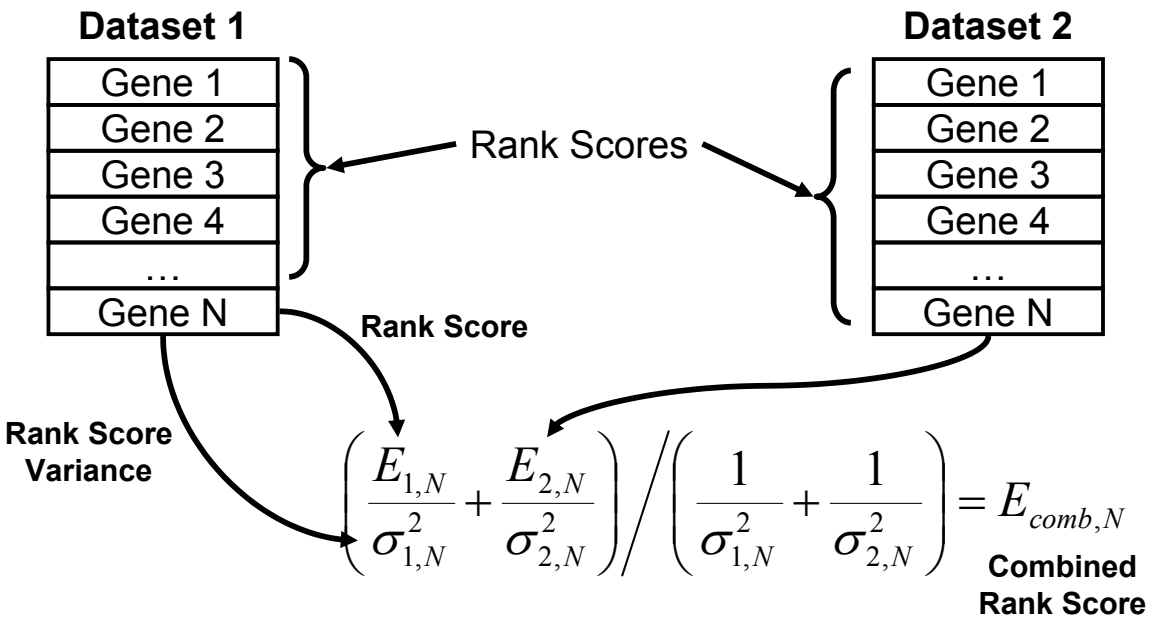


Figure 4: Data Combination Method. Estimated gene scores are combined using a weighted average. Weights are proportional to the standard deviation of the estimate.

genes [93]. This method is based on a simple scenario that compares the ranks of two genes, one of which is known to be differentially expressed. Mukherjee *et al.* computes the probability that a given ranking algorithm correctly ranks the differentially expressed gene by ranking it more favorably. They extend this method to multiple genes and define the random variables T_0 and T_1, which represent the ranks of null and differentially expressed genes, respectively. Assuming that a higher rank number corresponds to a more differentially expressed gene, they explicitly compute the probability, $P(T_0 < T_1)$, which they call the Binary Selection Accuracy (BSA) [93]. This probability is equivalent to the area under the curve (AUC) of a receiver-operator characteristic (ROC) curve produced by classifying genes into either differentially expressed or null groups using a rank threshold [8].

Instead of comparing different ranking algorithms, we compare the ranks produced from individual datasets to those of the combined dataset. In order to compute the BSA, we need a set of reference genes that are known to be differentially expressed.

For both the renal cancer and prostate cancer datasets, we identify genes from literature that have also been validated with qRT-PCR, the most common method for validating differentially expressed genes [90]. When computing the ROC curves, we weight each gene by one minus its p-value. Thus, statistically significant genes have a larger contribution to the resulting AUC.

3.3 Results and Discussion

3.3.1 Bootstrap Meta Analysis

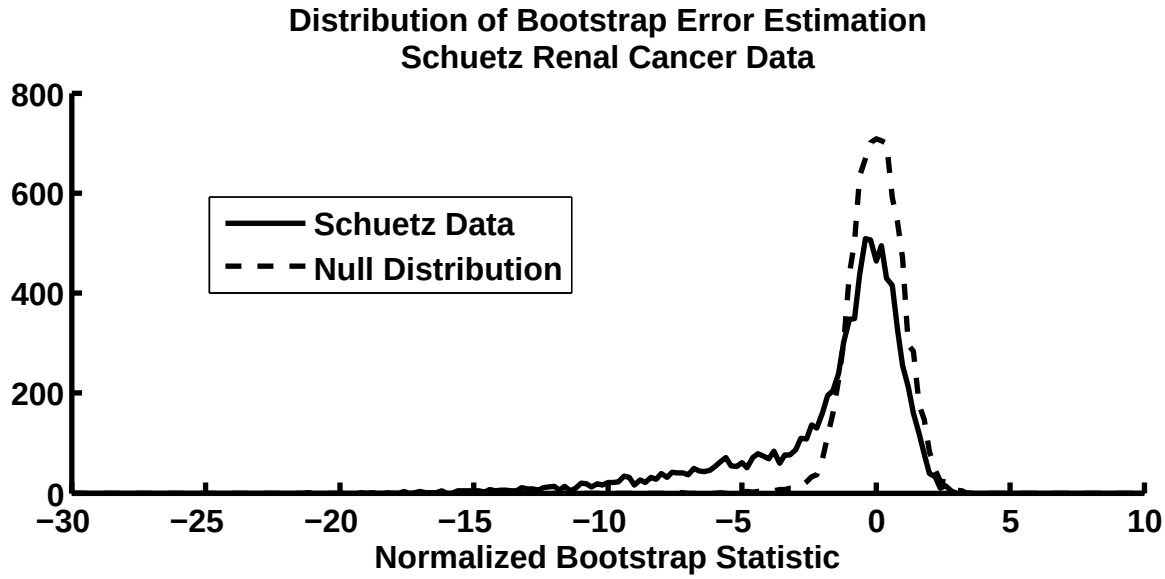
Data suggests that the no-information null distribution of the bootstrap is approximately normal, but the actual distribution of errors trained on correct class labels is skewed toward low error (**Figure 5**). This is expected, as the magnitude of skew should be a function of the separability of the classes under investigation and the information content of the genes present on the microarray.

For the prostate cancer datasets, this skew is less prevalent, as the empirical distribution more resembles the null distribution. Small deviations still exist at the low error side of the distribution, corresponding to potential differentially expressed genes (**Figure 6**).

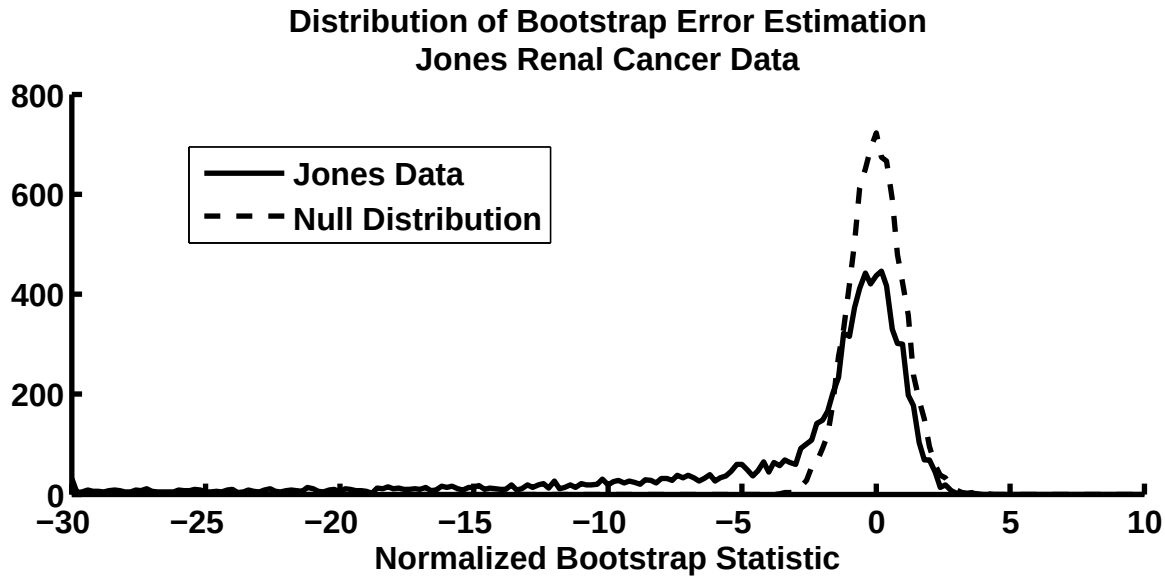
3.3.2 Validating the Relevance of Gene Ranking

We identify several validated genes for both the renal and prostate cancer datasets and use these genes as a reference for assessing the effect of data combination. We restrict these reference genes to those validated with qRT-PCR. This restriction improves the reliability of the reference genes since qRT-PCR is a common and established method for validating microarray assays [90].

In renal cancer, carbonic anhydrase IX (CA9) and low density lipoprotein-related protein 2 (LRP2) are up-regulated in CC compared to the ONC and CHR renal cancer subtypes [124, 22]. Additionally chloride channel Kb (CLCNKB), defensin beta 1, and parvalbumin (PVALB) are up-regulated in ONC and CHR compared to

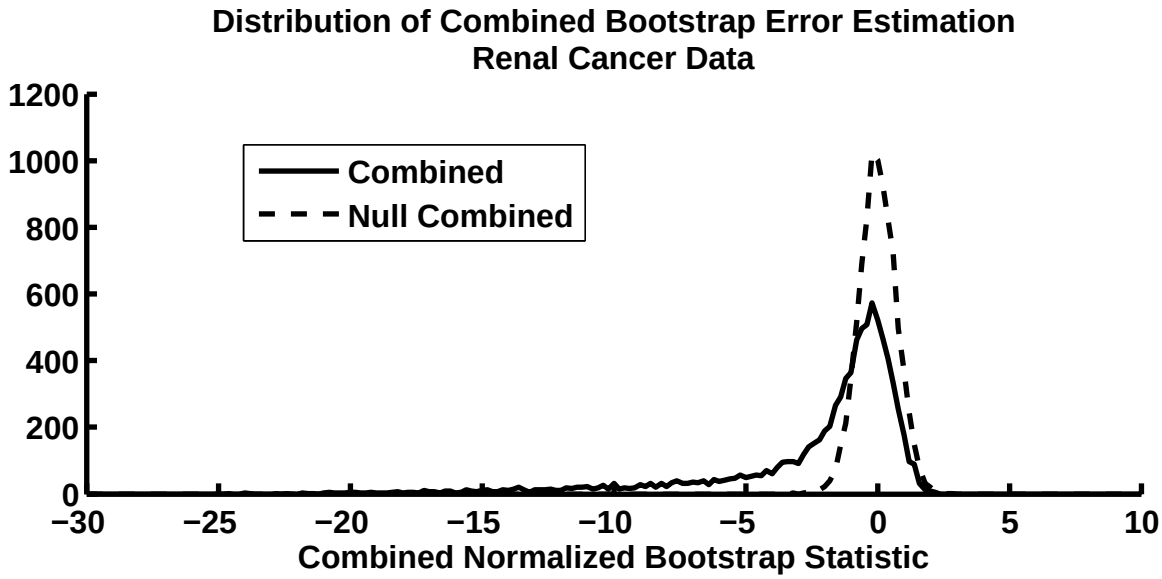


(a) Distribution of Schuetz renal cancer bootstrap errors.



(b) Distribution of Jones renal cancer bootstrap errors.

Figure 5: Distributions of normalized classification errors for the renal cancer datasets. Solid lines represent individual or combined distributions while dashed lines represent null distributions. Both the Schuetz (**5(a)**) and Jones (**5(b)**) datasets deviate significantly from the null distributions, indicating a large number of differentially expressed genes. This deviation is reflected in the combined data (**5(c)**, continued on the next page).



(c) Distribution of combined renal cancer bootstrap errors.

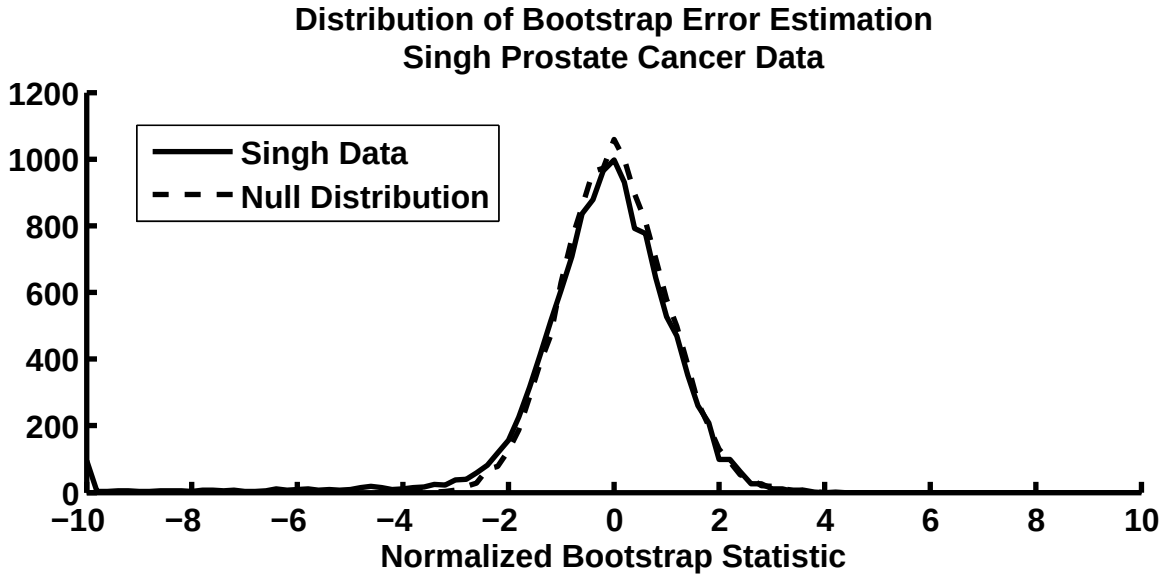
Figure 5 part (c). Parts (a) and (b) and full caption on the previous page.

Table 1: Validated renal cancer reference genes.

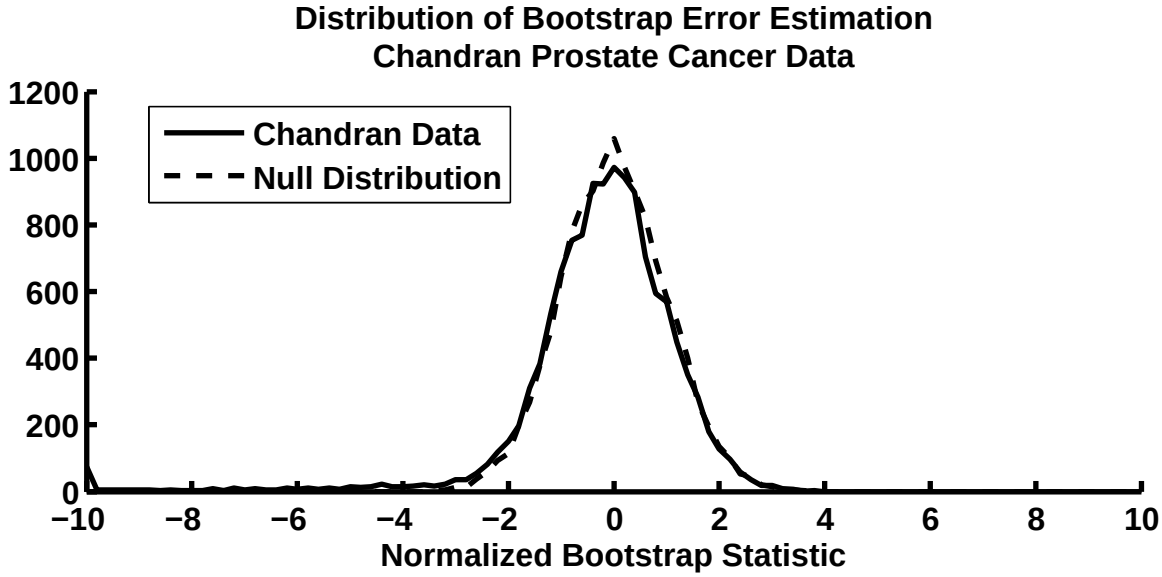
Gene	Regulation	Source
CA9	Up in CC	Chen 2005
CLCNKB	Up in ONC/CHR	Chen 2005
DEFB1	Up in ONC/CHR	Schuetz 2005
LRP2	Up in CC	Schuetz 2005
PVALB	Up in ONC/CHR	Chen 2005

Table 2: Validated prostate cancer reference genes.

Gene	Regulation	Source
AMACR	Up in Tumor	Ernst 2002
PLA2G7	Up in Tumor	Ernst 2002
HPN	Up in Tumor	Ernst 2002
PYCR1	Up in Tumor	Ernst 2002
SCGB1A1	Down in Tumor	Ernst 2002
TRIM29 (ATDC)	Down in Tumor	Ernst 2002

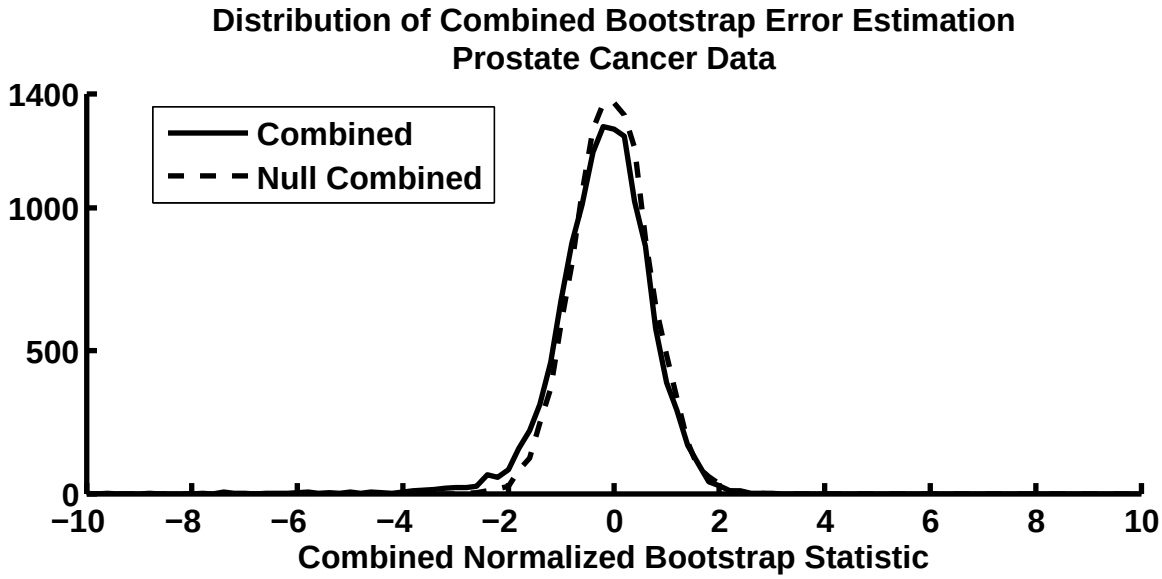


(a) Distribution of Singh prostate cancer bootstrap errors.



(b) Distribution of Chandran prostate cancer bootstrap errors.

Figure 6: Distributions of normalized classification errors for the prostate cancer datasets (Singh (**6(a)**), Chandran (**6(b)**), combined (**6(c)**, continued on the next page)). Compared to the renal cancer datasets, these datasets do not deviate significantly from the null distribution.

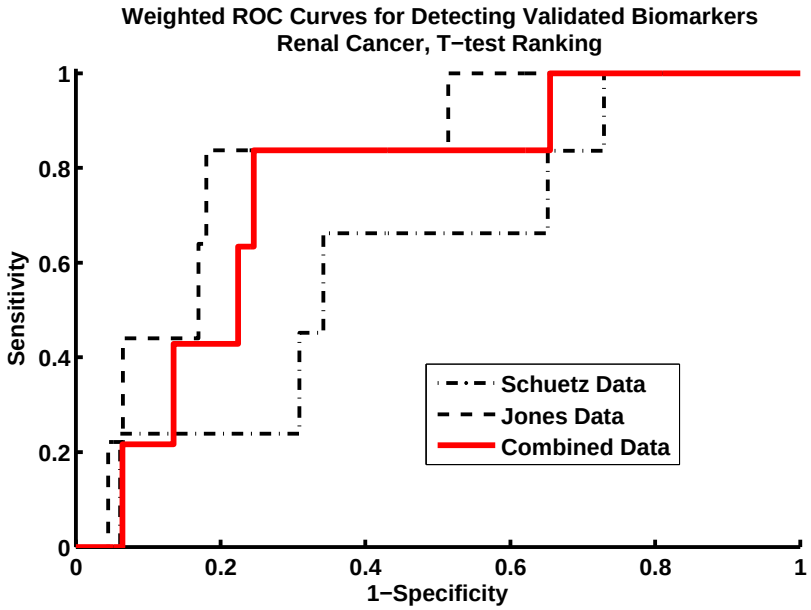


(c) Distribution of combined prostate cancer bootstrap errors.

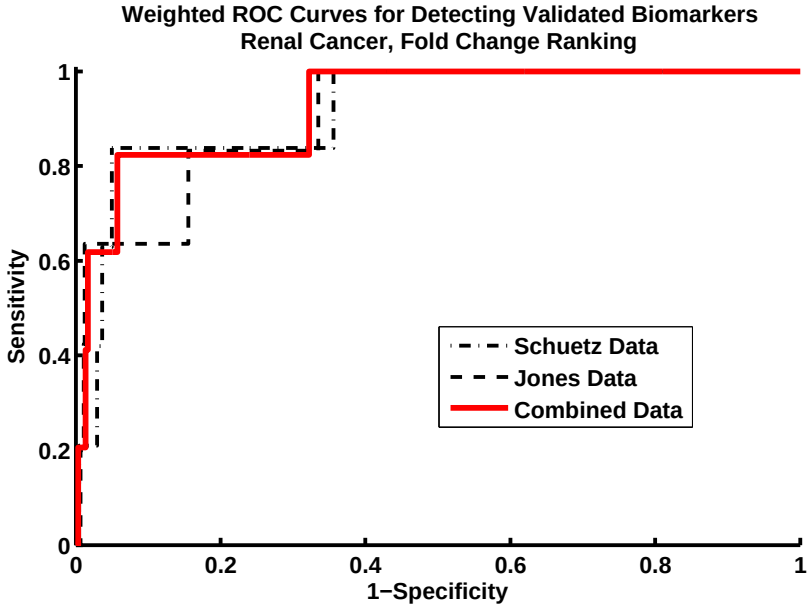
Figure 6 part (c). Figure parts (a) and (b) and full caption on the previous page.

CC (**Table 1**) [124, 22]. Using these reference genes, we compute the BSA (AUC) for each of the individual and combined renal datasets. We compare the results of the bootstrap meta-analysis to the original fold change combination method by Wang *et al.* as well as to a similar t-test method [147]. The ROC curves (**Figure 7**) for the combined data are more similar to the Jones dataset when using T-test or bootstrap ranking. ROC curves for the fold change method are similar between both individual datasets and the combined dataset. BSA values (**Figure 9**) for the combined dataset are higher than both of the individual datasets when ranking genes using fold change and bootstrap methods, but not the t-test method. Fold change tends to perform well for both individual datasets in terms of favorably ranking the reference genes. Thus, we also expect fold change to perform well for the combined data.

Compared to the Jones data, the small sample size of the Schuetz data seems to reduce the efficiency of ranking using the t-test and bootstrap methods. Small sample size generally corresponds to higher measurement variance. Therefore we expect the data combination method to reduce the overall contribution of the Schuetz

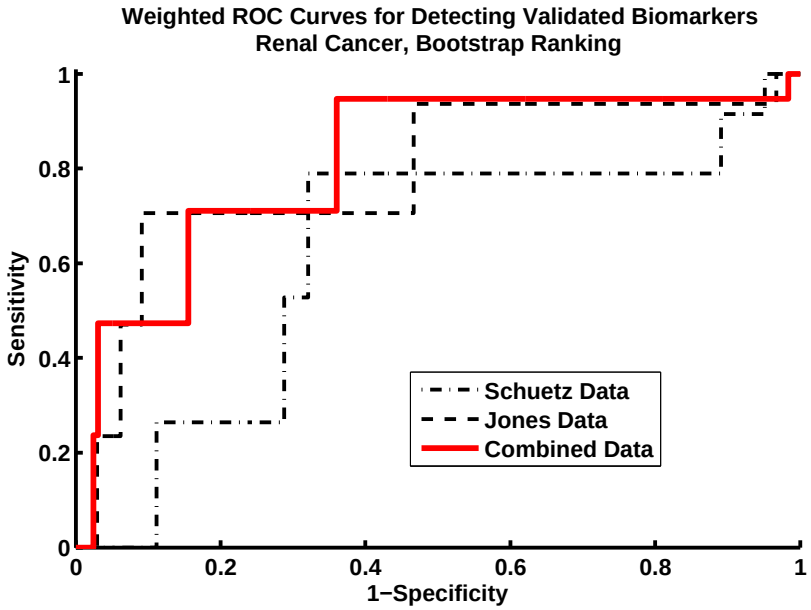


(a) ROC curves for detecting reference renal cancer genes after t-test data combination.



(b) ROC curves for detecting reference renal cancer genes after fold-change data combination.

Figure 7: ROC curves for detecting validated reference genes for renal cancer. Red lines are combined data and dashed lines are individual datasets. For all datasets, fold change (**7(b)**) tends to detect reference genes more efficiently compared to t-test (**7(a)**) and bootstrap (**7(c)**, continued on the next page). Combining data using fold change and bootstrap slightly improves detection efficiency. This corresponds to an increase in area under the ROC curve.

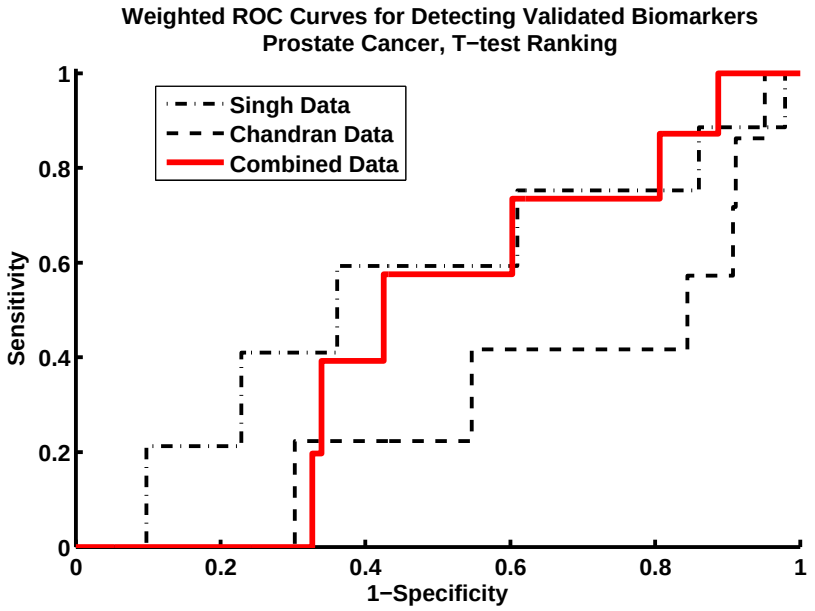


(c) ROC curves for detecting reference renal cancer genes after bootstrap data combination.

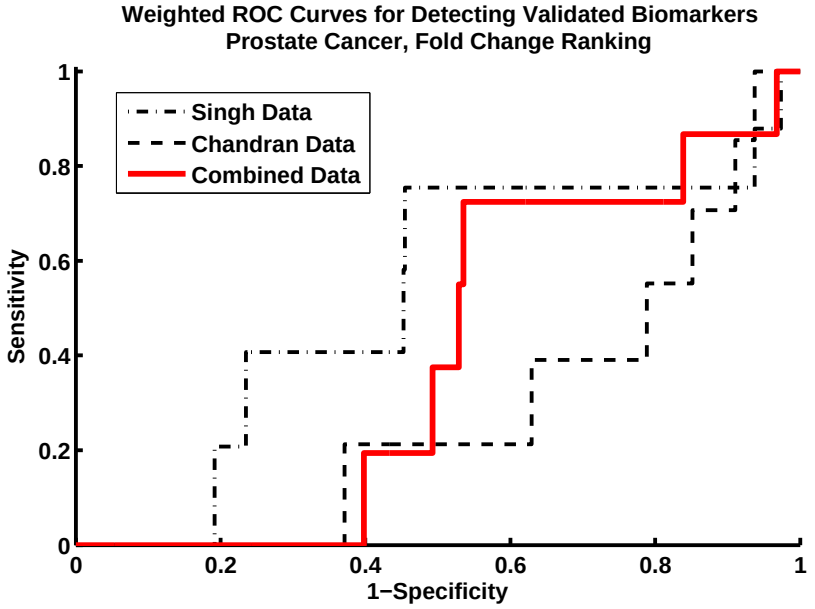
Figure 7 part (c). Figure parts (a) and (b) and full caption on the previous page.

data. However, the BSA for the t-test combination method is higher than that of the individual Schuetz dataset and lower than that of the individual Jones data. This suggests that, for the t-test method, the contribution of the Schuetz data to the combined data ranking has not been properly weighted to account for its higher variance. The results of bootstrap ranking using individual datasets produces similar BSA values compared to the t-test. However, our bootstrap method properly weights the Schuetz data to reduce its overall contribution in the combined data. Thus, the BSA of the combined data ranking is higher than both individual datasets (**Figure 9**).

For the prostate cancer data, we select six validated genes from literature, four of which are over-expressed in tumor tissue while two are under-expressed (**Table 2**) [39]. We compute the ROC curves and BSA scores for the prostate cancer datasets using these reference genes (**Figure 8**, **Figure 10**). Compared to the renal cancer data, detection of these reference genes via ranking is much less efficient, resulting relatively

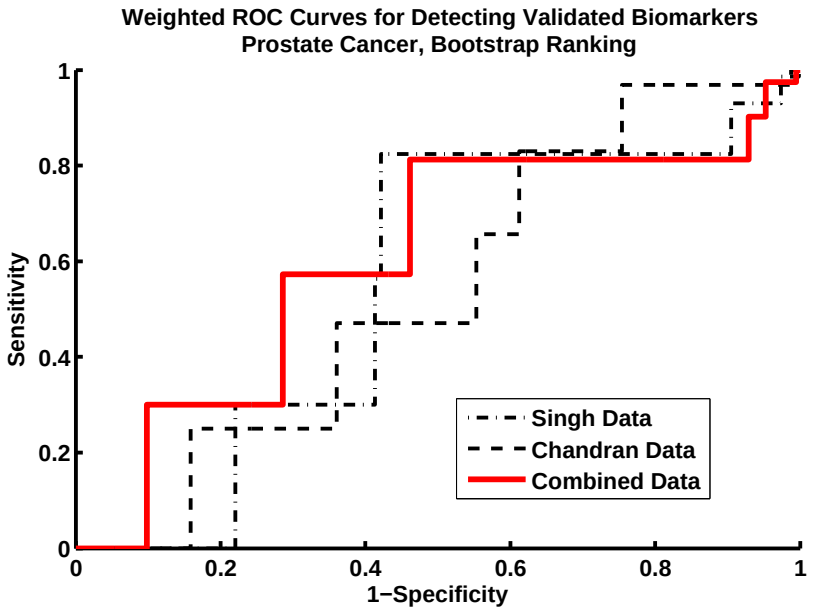


(a) ROC curves for detecting reference prostate cancer genes after t-test data combination.



(b) ROC curves for detecting reference prostate cancer genes after fold-change data combination.

Figure 8: ROC curves for detecting validated reference genes for prostate cancer. Red lines are combined data and dashed lines are individual datasets. Combined data does not improve efficiency of reference gene detection when using the t-test (**8(a)**) or fold change (**8(b)**) methods. The bootstrap method (**8(c)**, continued on the next page) slightly increases detection performance of the reference genes for combined data.



(c) ROC curves for detecting reference prostate cancer genes after bootstrap data combination.

Figure 8 part (c). Figure parts (a) and (b) and full caption on the previous page.

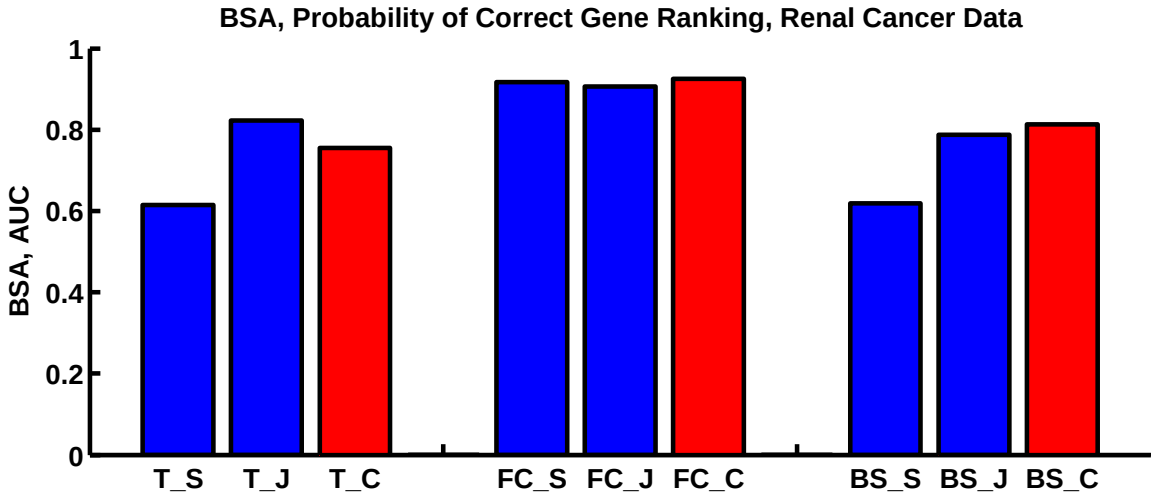


Figure 9: BSA (AUCs) of individual and combined renal cancer datasets for detecting reference genes. The red bars are AUCs of the combined data. T, FC, and BS correspond to t-test, fold change, and bootstrap, respectively. S, J, and C correspond to Schuetz, Jones, and combined data, respectively. For the fold change and bootstrap (middle and right bar triplets), the relevance of ranking for the combined data is at least as good, if not better, than both individual datasets.

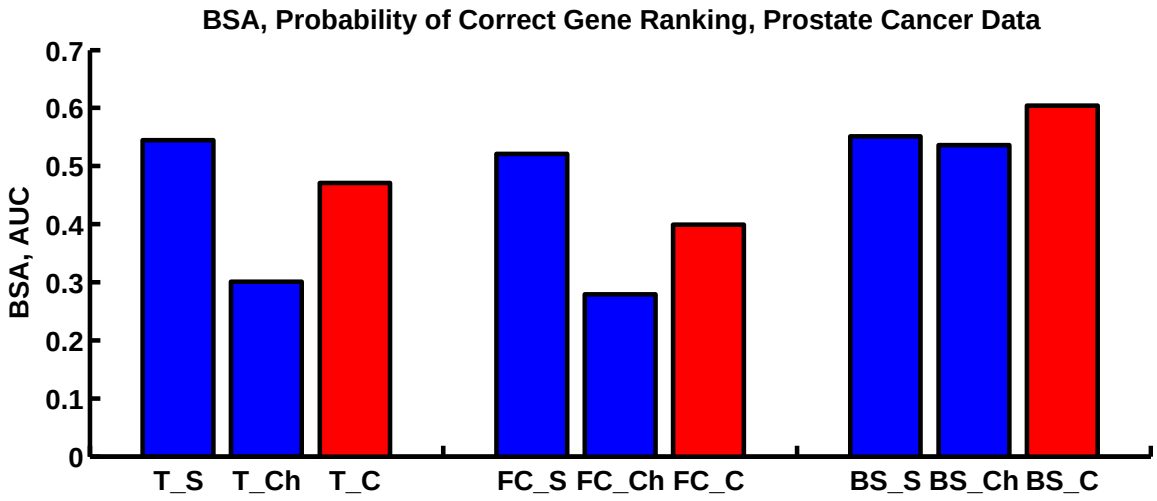


Figure 10: BSA (AUCs) of individual and combined prostate cancer datasets for detecting reference genes. The red bars are AUCs of the combined data. T, FC, and BS correspond to t-test, fold change, and bootstrap, respectively. S, Ch, and C correspond to Singh, Chandran, and combined data, respectively. The bootstrap combination method (right triplet of bars) outperforms both the t-test and fold change methods.

low BSA scores. This decrease in efficiency may be due to biological heterogeneity among the three datasets from Singh, Chandran, and Ernst [133, 19, 39].

Despite the low BSA scores, the genes identified from the Ernst dataset have been qRT-PCR validated and should serve as a point of reference for assessing the quality of the Singh and Chandran datasets. Examining the ROC curves and the corresponding BSA values, for both the t-test and fold change data combination methods, we see that combining data improves gene detection compared to the individual Chandran dataset. However, for these ranking methods, the combined dataset still performs worse than the individual Singh dataset. The bootstrap method performs equally well on both the individual Singh and Chandran datasets. Furthermore, the combined method improves the overall performance of gene detection compared to both individual datasets.

3.3.3 Interpretation of Selected Genes

Data combination using bootstrap meta-analysis improves the ranks of two of the prostate cancer reference genes. Both of these genes, tripartite motif containing 29 (TRIM29) and pyrroline-5-carboxylate reductase 1 (PYCR1), were previously validated by Ernst *et al.* [39]. For TRIM29, individual dataset p-values of 0.07 and 0.05 are reduced to 0.02 in the combined data. Likewise, for PYCR1, individual dataset p-values of 0.19 and 0.16 are reduced to 0.10 in the combined data. The ranks for the other four reference genes are not improved in the combined data compared to both individual datasets. However, the individual dataset p-values of these four genes are larger than those of TRIM29 and PYCR1. Although these genes have been validated with qRT-PCR, their reliability for these particular datasets is questionable and provides further evidence of the analytical difficulties due to prostate tissue heterogeneity.

In addition to the reference genes, we identify several other prostate cancer-related genes using the bootstrap-combined data. Interestingly, brain-derived neurotrophic factor (BDNF), has been linked to prostate cancer [13]. We easily identify BDNF in the Singh dataset (rank=3), but not in the Chandran data (rank=420). The combined data ranks the gene at 17 (lower rank is better). Similarly, the combined data improves the recognition of the metastasis-associated protein 1 (MTA1, combined rank=18, Singh rank=16, Chandran rank=360) and chemokine receptor 2 (CCR2, combined rank=35, Singh rank=5, Chandran rank=580), again by discounting the contribution of the Chandran data [55, 83]. The combined data also favorably ranks some relevant genes that are not ranked favorably in individual datasets. For example, ITIH3 (Singh rank=25, Chandran rank=326, combined rank=6) and CHD5 (Singh rank=165, Chandran rank=95, combined rank=7) have both been linked to human cancer [51, 3].

Data combination also improves the ranks of two of the renal cancer reference

genes, CLCNKB and PVALB. The rank of DEFB1 is only improved compared to the individual Schuetz dataset (Schuetz rank=6158, Jones rank=552, combined data=1011). However, the p-value of DEFB1 in the Schuetz dataset is large. Likewise, the combined data rank of CA9 is improved compared to the Jones dataset, but not the Schuetz dataset (Schuetz rank=1900, Jones rank=2808, combined data=2344). Some genes implicated in renal cancer are favorably ranked in the combined data, but not in either of the individual datasets. For example, CXCR4 has a combined data rank of 28, but individual ranks of 113 and 188 in the Schuetz and Jones dataset, respectively [123]. Many other examples of improved biological relevance exist in both prostate and renal cancer datasets. However, a more rigorous biological analysis is beyond the scope of this paper.

3.4 Conclusion

Until microarray technology becomes standardized, we must develop statistical methods to handle small sample data. Standardization protocols are currently not well defined. However, the ideal scenario for standardization should allow us to compare quantitative measurements across different platforms or to expect all clinical measurements to be acquired with the same technology. Regardless of standardization, microarrays are still subject to technological variance when experiments are performed at different times or locations. Our proposed method is a possible solution to reduce technical bias by computing differential gene expression scores on distinct microarray groups, then combining these scores across multiple groups of microarrays. The bootstrap accurately estimates classification errors for genes of individual small datasets while the combination method favors genes whose scores have lower individual dataset variances. The results of this method applied to prostate and renal cancer datasets indicate that bootstrap meta-analysis improves the biological relevance of gene selection by increasing data sample size.

CHAPTER IV

IMPROVING THE EFFICIENCY OF BIOMARKER IDENTIFICATION USING BIOLOGICAL KNOWLEDGE

4.1 Introduction

The subjective nature of traditional medical techniques limits the accuracy of cancer subtype classification and, subsequently, the effectiveness of therapy. Clinicians visually examine cancer specimens to determine their subtypes before proposing treatment regimens. However, cancers with similar characteristics may behave very differently despite similar treatment conditions [50]. Because cancer is the result of genetic anomalies, emerging diagnostic research has primarily focused on genetic and proteomic expression. This research generally involves the use of high throughput technology (e.g. microarrays and mass spectrometry) to generate large amounts of genetic and proteomic expression data. We typically reduce this data using one of many analysis algorithms with the goal of identifying a subset of features (corresponding to genes or proteins) with high predictive accuracy [143, 124, 133]. These feature subsets, if correctly identified, will both enhance our understanding of the biological mechanisms as well as provide us with an accurate diagnostic system. When validated, we call these differentially expressed features biomarkers. Unfortunately, even the selection of a ranking metric is subjective, as different metrics may identify different subsets of features [10]. Feature ranking affects both the efficiency of identifying relevant genes and the accuracy of subsequent predictive models. We address this issue by presenting a method that uses existing biological knowledge to identify the best feature ranking metric for a particular gene expression dataset. The optimal metric maximizes the probability of correctly ranking differentially expressed

and previously validated genes (**Figure 11**). **Figure 11** illustrates a hypothetical situation in which prior knowledge enables us to choose between two feature ranking algorithms. Although this example ranks genes individually, we can generalize it to combinatorial feature selection. In **Figure 11**, we assume that genes 8, 52, and 234 have been previously identified and validated for a particular clinical problem. For example, if the clinical problem is prostate cancer recurrence, we want to identify genes from various knowledge sources that are differentially expressed between event-free survival (survival without symptoms) after a number of years and cancer recurrence. Among the multiple feature ranking metrics, the "optimal" or biologically relevant metric should favorably rank these previously validated genes while simultaneously reducing the number of false discoveries. The chosen ranking metric is only optimal within the space of tested ranking metrics and available knowledge.

Despite numerous feature selection studies in the literature, there is still a lack of clinically validated and proven biomarkers for many cancers. Thus, the use of "correct" genes as knowledge for algorithm selection is subjective and we should choose these genes carefully. Sources of biological knowledge are abundant, but vary in terms of reliability. We consider a knowledge source to be reliable if genes (or the corresponding expressed proteins) from that source have been clinically validated as differentially expressed. The majority of knowledge is contained in the literature and we can roughly divide them into four levels of reliability, adapted from a review of post-analysis validation methods by Chuaqui *et al.* [25]:

1. **No biological validation.** As the lowest level of reliability, this includes studies that develop feature selection algorithms and present the selected list of genes without a stringent interpretation of the biological results.

2. ***In silico* validation.** Also known as computational validation, these studies compare their feature selection results to the results of other studies. They may

also identify Gene Ontology (GO) categories that are statistically overrepresented as a result of feature selection.

3. **Same-sample validation.** These studies validate their microarray experiments by performing additional assays on the same samples from which their microarrays were derived. These assays typically include quantitative real-time PCR (qRT-PCR) or northern analysis and serve to validate the technical reliability of the microarrays.

4. **Independent or clinical validation.** As the highest level of reliability, these studies validate the results of their microarray experiments using independent biological samples, usually from a clinical source. Independent validation ensures that the selected features are not a result of over-fitting. These validations often take the form of qRT-PCR and in situ hybridization (ISH) for RNA products, or immunohistochemistry (IHC) and western analysis for protein products.

Despite frequent disagreement between qRT-PCR and microarray results, qRT-PCR is the most common method for validation of differentially expressed genes. Genes with large fold-change in microarray data are consistently correlated with qRT-PCR while those with smaller fold change are more susceptible to technical variability [90]. The detection of differentially expressed genes is generally reproducible across several microarray platforms [126]. However, in light of a recent study illustrating the pervasiveness of technical artifacts in microarray data [137], we only consider a knowledge source reliable if it falls into category three or four.

Investigators have attempted to improve feature selection by using biological knowledge. Their knowledge sources often fall into category two of reliability, in silico validation, and include Gene Ontology and pathway databases, published literature, microarray repositories, and sequence information. Generally, these studies identify genes that cluster or correlate with genes from the knowledge sources [1, 72, 71].

Another study developed a theoretical framework to compare feature ranking metrics in the presence of control features [93]. However, this study also neglected to focus on the reliability of the control features. Indeed, the wealth of available information in the form of gene and protein interactions, functional annotation, and genetic and pathways can improve the results of data analysis. Furthermore, microarray data analysis has shifted from purely data driven methods to methods that use additional knowledge, even in the feature selection process [5].

We develop a method to quantify the efficiency of detecting biomarkers by feature ranking. This method maximizes the biological relevance of feature ranking by choosing the best metric from a population of metrics. The chosen ranking metric is optimal with respect to knowledge obtained from reliable sources. We test the effectiveness of our method using clinical gene expression data. Results indicate that the choice of ranking metric significantly affects feature ranking, which, in turn, affects the efficiency of discovering and validating novel biomarkers.

4.2 Methods

4.2.1 Gene Ranking and Selection

Throughout this section, the term 'gene set' denotes a group of one or more genes that act in concert. A 'sample' refers to measurements of a gene set from a single microarray or molecular profile. The entire microarray sample contains ℓ genes while a gene set may contain p genes (where $p \ll \ell$). We represent a gene set i for sample n as vectors, $\vec{x}_n^i \in \mathbb{R}^p$, and labels, $y_n \in \{0, 1\}$. The class label, y_n, indicates the clinical source of the microarray sample. In most cancer problems, $y_n = 1$ indicates, for example, samples measured from patients with cancer and $y_n = 0$ indicates samples from patients with no cancer. For a microarray dataset with N samples, gene set i for a particular dataset is the vector $\vec{d_i} = ((y_1, \vec{x}_1^i), (y_2, \vec{x}_2^i), \ldots, (y_N, \vec{x}_N^i))$.

For each gene set, we assign a score that represents the predictive ability of that

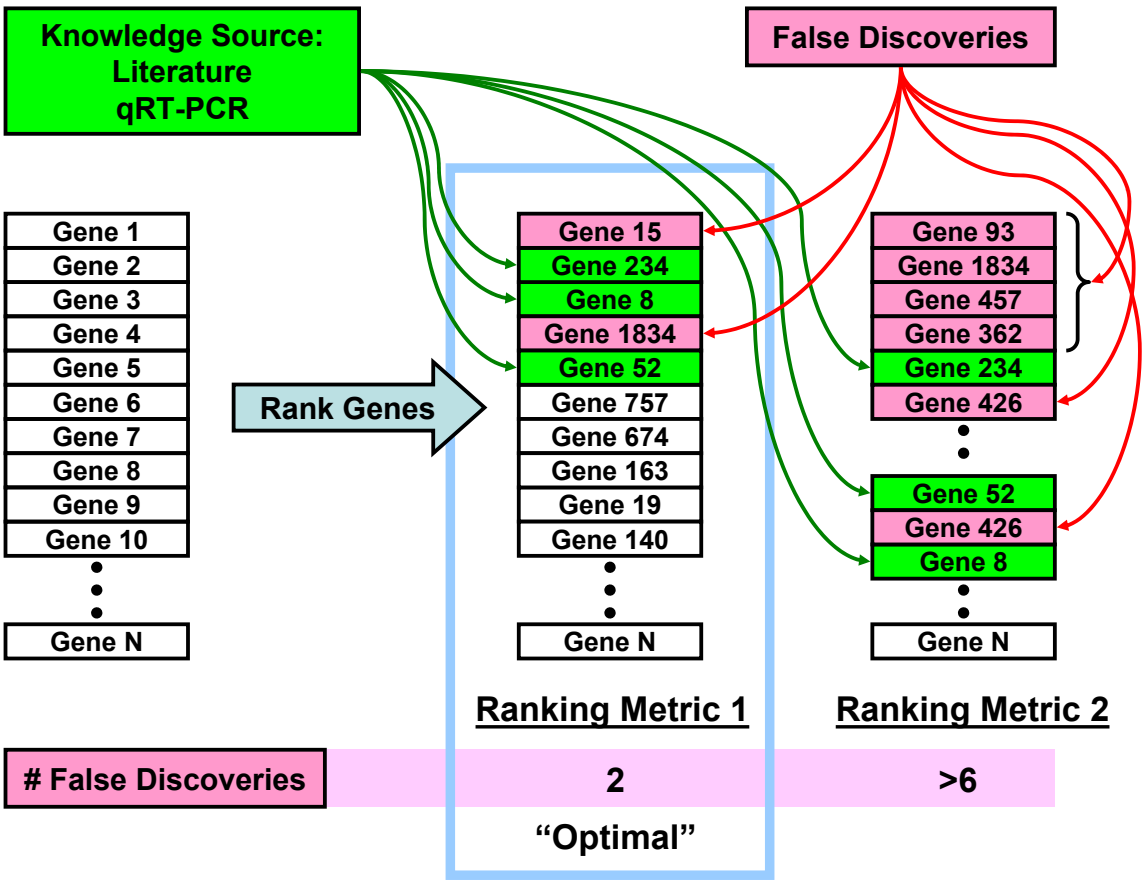


Figure 11: Selection of a biologically relevant ranking metric using existing biological knowledge. The "optimal" method–ranking metric 1–minimizes the number of false discoveries. False discoveries are genes that have not been validated or that have been shown to be non-informative.

Table 3: Parameters for wrapper-based feature selection methods.

Classifier	Kernel	Parameters
SVM	Linear	C:0.01,0.04,0.07,0.1,0.4,0.7,1,4,7,10,40,70,100
	Gaussian	C:0.01,0.04,0.07,0.1,0.4,0.7,1,4,7,10,40,70,100 γ:0.001,0.004,0.007,0.01,0.04,0.07,0.1,0.4,0.7, 1,4,7,10,40,70,100
SDF	Linear	N/A
	Gaussian	γ:0.001,0.004,0.007,0.01,0.04,0.07,0.1,0.4,0.7, 1,4,7,10,40,70,100
LDA	Linear	N/A
	Gaussian	γ:0.001,0.004,0.007,0.01,0.04,0.07,0.1,0.4,0.7, 1,4,7,10,40,70,100

gene set using the function

$$\alpha_i = h_\theta(\vec{d_i}) \tag{11}$$

where $\theta \in \Theta$ is a meta-parameter that characterizes the scoring function, or ranking metric. A smaller α indicates a more differentially expressed gene set. We also constrain α to be in the interval $[0, 1]$, such that the function h_θ may be a hypothesis test that computes a p-value or an estimate of classification error. Although Θ can represent the space of all ranking methods, we use a reduced set of wrapper-based and filter methods in our simulations. Specifically, we use three classifiers–support vector machines (SVM) [28], signed distance function (SDF) [6], and linear discriminant (LDA)–with various parameters and the common t-test, fold change, and significance analysis of microarrays (SAM) [141]. Refer to the Appendix for more details about each classifier. We use the 0.632+ bootstrap to estimate classification error for the wrapper methods [10, 37]. We discretely vary the classifier parameters over several values (**Table 3**).

In practice, a gene expression dataset will have N samples, each with ℓ features. We separately examine m feature sets (m can be different from ℓ and include, for example, all pairs, triplets, or a subset of feature combinations), corresponding to

$(\vec{d_1}, \vec{d_2}, \ldots, \vec{d_m})$. From the mapping defined in **Equation 11**, we compute the set of rank scores for each feature set $(\alpha_1, \alpha_2, \ldots, \alpha_m)$. Using a simple selection method, we can then conclude that the best feature sets and potential biomarkers are in the set

$$G = \{i : \alpha < \tau\} \tag{12}$$

where τ is a threshold.

4.2.2 Selection of a Ranking Metric Using Maximum Likelihood

We want to choose a feature set ranking metric, θ, that produces the most biologically relevant ranking of the m feature sets, $(\vec{d_1}, \vec{d_2}, \ldots, \vec{d_m})$, with respect to a given set of knowledge. Although we may never know the biological relevance of all features in a dataset, we may infer from literature that the k feature sets, $G_k = \{g_1, g_2, \ldots, g_k\}$, are relevant, where $k \ll m$ and the elements of the set correspond to the feature sets $(\vec{d}_{g_1}, \vec{d}_{g_2}, \ldots, \vec{d}_{g_k})$. Assuming that lower scores are better, the best θ assigns scores such that $h_\theta(\vec{d_i}) < h_\theta(\vec{d_j})$ for $i \in G_k$ and $j \neq G_k$, i.e., feature set i is known to be more biologically relevant than feature set j for this particular dataset. This implies that the elements of the set $\{h_\theta(\vec{d_i}) : i \in G_k\}$ should generally be smaller than those of $\{h_\theta(\vec{d_j}) : j \notin G_k\}$. If the knowledge is reliable, we want to choose a θ that maximizes the probability that the score of a feature set from G_k is less than that of a feature set that is not from G_k. Explicitly, we define this probability as

$$\phi(G_k, \theta) = P\left(h_\theta(\vec{d_i}) < h_\theta(\vec{d_j})\right) \tag{13}$$

for $i \in G_k$ and $j \notin G_k$. This probability represent the biological relevance score of θ, and can be approximated from the data using

$$\phi(G_k, \theta) = \frac{1}{k(m-k)} \sum_{i \in G_k} \sum_{j \notin G_k} I\left(h_\theta(\vec{d_i}) < h_\theta(\vec{d_j})\right) \tag{14}$$

where $I(x)$ is the indicator function that evaluates to one when x is true and zero when x is false. **Equation 14** is equivalent to computing the area under an ROC

curve (AUC) for classifying feature sets as either biologically relevant or irrelevant [93]. Using this biological relevance function, we can identify the optimal ranking metric, θ, using maximum likelihood estimation:

$$\hat{\theta} = \underset{\theta}{\operatorname{argmax}}\, \phi(G_k, \theta). \tag{15}$$

For a more detailed derivation of maximum likelihood ranking metric selection, refer to **Appendix A.1**.

4.2.3 Selection of a Ranking Metric Using Maximum *A Posteriori*

The maximum likelihood approach to identifying an optimal $\hat{\theta}$ takes all current information about known feature sets into account when computing the likelihood. Thus, when we introduce new knowledge, we can combine this new knowledge with the existing knowledge to recompute the likelihood. However, when introducing new knowledge, we may also use the probability from a previous knowledge set as a prior within a Bayesian framework.

For example, suppose that we know, from previous experiments, that some feature selection methods tend to perform better than others for particular datasets. We can quantitatively represent this as prior knowledge, $P(\theta)$. If the previous experiments have resulted in a set of k validated genes, then we can use an explicit formula to approximate a prior:

$$P(\theta) = \frac{\phi(G_k, \theta)}{\sum_{\theta' \in \Theta} \phi(G_k, \theta')} \tag{16}$$

Given only the prior, we can estimate the optimal algorithm in the same manner as the maximum likelihood method. However, suppose that we are given additional information about n biologically relevant feature sets, $G\prime_n$. Then we can update our beliefs about the feature ranking metrics using the formula for computing a posterior distribution and use the maximum *a posteriori* (MAP) method to estimate $\hat{\theta}$:

$$\hat{\theta} = \underset{\theta}{\operatorname{argmax}}\, \phi(G'_n, \theta) P(\theta). \tag{17}$$

For a more detailed derivation of the maximum *a posteriori* method, refer to **Appendix A.2**.

4.2.4 Iteratively Updating Knowledge

The purpose of identifying $\hat{\theta}$ is to use this "optimal" feature ranking metric to identify and validate new biomarkers. augment our knowledge by identifying new gene sets that may also be relevant. The optimal $\hat{\theta}$ helps us identify gene sets that are most likely to be biologically relevant, increasing the overall efficiency of data mining.

It may be difficult to compile a comprehensive list of knowledge from literature and independent validation. Consequently, we can expect that some feature sets that are not in our knowledge set, $\{i \notin G_k\}$, are, in fact, relevant biomarkers. We define V as the set of all relevant biomarkers, regardless of whether their relevance is known. A feature set is known to be in the set V only after performing a validation procedure such as qRT-PCR. **Figure 12** and **Figure 13** are descriptions of the knowledge update algorithms for both the ML and MAP methods. Refer to **Appendix A.1** and **A.2** for further details.

initialize knowledge set G_k
while $G_k \neq V$ do
    $\hat{\theta} = \underset{\theta}{\operatorname{argmax}}\, \phi(G_k, \theta)$
    identify and validate n new feature sets: G'_n
    update knowledge set: $G_{k+n} = \{G_k, G'_n\}$
    $k = k + n$
end while

Figure 12: Iteratively updating knowledge using maximum likelihood.

initialize knowledge set G_k

initialize prior $P(\theta) = \phi(G_k, \theta) / \left[\sum_{\theta' \in \Theta} \phi(G_k, \theta') \right]$

$\hat{\theta} = \underset{\theta}{\operatorname{argmax}} P(\theta)$

while $G_k \neq V$ do

    identify and validate n new feature sets: G'_n

    $\hat{\theta} = \underset{\theta}{\operatorname{argmax}} \phi(G'_n, \theta) P(\theta)$

    update knowledge set: $G_{k+n} = \{G_k, G'_n\}$

    update prior: $P(\theta) = P(\theta | G'_n)$

    $k = k + n$

end while

Figure 13: Iteratively updating knowledge using maximum *a posteriori*.

4.2.5 Assessing the Efficiency of a Ranking Metric

If we know all feature sets in the set V, we can quantify any improvement in efficiency due to optimization of the ranking metric. Using bootstrap resampling, we randomly and repeatedly partition the feature sets in V into a group of known relevant feature sets (training) and a group of unknown relevant feature sets (testing). If there are K elements in V, we randomly select K elements with replacement, resulting in K^* ($K^* < K$) unique elements for the testing set. We use the group of $K - K^*$ known relevant feature sets to optimize the ranking metric, then iteratively detect feature sets from the unknown set of K^* features and update our knowledge set. Every validation test requires a finite amount of time and resources. Plotting the fraction of correctly validated biomarkers (y-axis) vs. total validation time (x-axis), reveals that higher detection efficiency corresponds to a larger area under this curve. This curve is similar to a ROC curve, so we also call the area under this curve the AUC. We repeat this bootstrap sampling of feature sets 100 times in order to compute the significance of the differences among three conditions: optimal metric selection, sub-optimal metric selection, and sub-optimal initial knowledge. For the sub-optimal metric selection condition, we use correct initial knowledge selected from the set V

via bootstrap, but use a modified equation to choose $\hat{\theta}$ with median AUC rather than maximum:

$$\hat{\theta} = \underset{\theta}{\text{argmedian}}\, \phi(G_k, \theta) \tag{18}$$

or using the Bayesian maximum *a posteriori* approach:

$$\hat{\theta} = \underset{\theta}{\text{argmedian}}\, \phi(G_k, \theta) P(\theta) \tag{19}$$

Selection of a ranking metric with median AUC represents the practice of arbitrarily selecting a metric with no regard for biological relevance and efficiency. This median AUC algorithm also serves as a reference point for assessing the potential improvement of efficiency when using the optimal algorithm. For simulations using clinical data, we directly compare the optimal selection method using maximum likelihood or maximum *a posteriori* to a commonly used filter method such as fold change or significance analysis of microarrays (SAM) [141].

For the sub-optimal initial knowledge condition, we begin the simulation with incorrect knowledge selected via bootstrap and use the maximum likelihood or maximum *a posteriori* method to optimize the ranking algorithm before updating the current knowledge set. We expect the average AUC of the optimal selection condition to be higher than that of both of the sub-optimal conditions. **Figure 14** illustrates this process.

To determine whether the optimization procedure is over-fitting to the knowledge set, we conduct additional tests using randomly selected knowledge sets. If over-fitting is occurring, results of the optimal, suboptimal, and suboptimal knowledge tests for randomly selected knowledge should be similar to those of the true knowledge set.

4.2.6 Synthetic Data Simulations

Synthetic gene expression datasets are often used to assess the efficacy of feature selection algorithms [89, 94, 82]. Clinical gene expression datasets usually suffer from small sample size and, as a result, we cannot accurately estimate properties such as

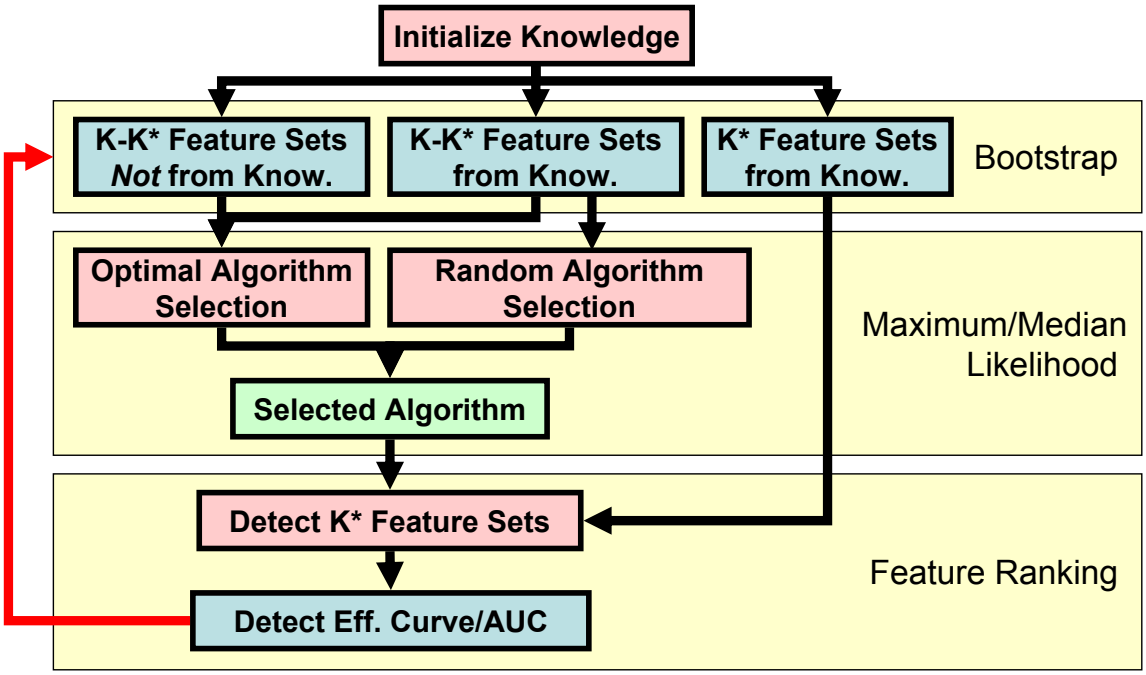


Figure 14: Quantifying the efficiency of detecting relevant biomarkers. For clinical data, we define an initial set of knowledge with K known differentially expressed feature sets. Using bootstrap cross validation, we partition the knowledge set into K^* and $K - K^*$ samples. K^* is the number of unique samples after sampling from the knowledge set K times with replacement. We optimize the ranking algorithm using $K - K^*$ feature sets and assess the algorithm's efficiency in detecting the remaining K^* feature sets. For each of the three conditions–optimal metric selection, sub-optimal metric selection, and sub-optimal initial knowledge–we perform this bootstrap sampling 100 times in order to compute the significance of any differences between mean AUC values.

noise and differential expression until after fully validating the results with external data. Synthetic datasets allow us to control these properties, replicate experiments, and compute a confidence interval for the analysis results. One of the caveats of synthetic data generation is that they must closely resemble true datasets in terms of gene expression distributions, sample size, and number of features.

In the following synthetic data simulations we use controlled experiments to verify that using optimally selected ranking metrics improves the efficiency of feature selection. It is difficult to gauge improvement of feature selection efficiency in clinical datasets because the distribution of data is unknown. Synthetic datasets for these simulations have variable sample sizes and consist of 3000 one- or two-dimensional feature sets. We use a small dataset size of 20 samples divided into two classes (10 samples per class) and a larger dataset size of 40 samples (20 samples per class). One hundred of these feature sets are distributed such that samples are differentially expressed between the two classes. The remaining 2900 feature sets are distributed with no differential expression (**Table 4**, one-dimensional feature sets and **Table 5**, two-dimensional feature sets). In all synthetic datasets, the knowledge set, V, contains 100 differentially expressed feature sets. We model synthetic datasets to be roughly similar to real expression data in terms of both number of feature sets and sample size. The number of differentially expressed feature sets is small compared to the total number of feature sets.

We use six synthetic datasets to examine feature set detection efficiency (**Table 6**). The first and second datasets use either linearly or non-linearly separable biomarkers for feature sets 1 to 100. The third dataset uses a mixture of both linearly and non-linearly separable biomarkers. Each of these datasets is also varied by increasing noise level. For example, the datasets with linearly separable noise have either 10% or 30% overlap between class 1 and class 2 distributions (**Figure 15**, rows one and

Table 4: One-dimensional Gaussian distribution means for relevant and irrelevant feature sets in the synthetic datasets. Standard deviations of Gaussian distributions are one.

Group Label	Feature Set Group	Class 1 Mean	Class 2 Mean
A	Diff. Exp. Linear 10% Overlap	-1.2816	1.2816
B	Diff. Exp. Linear 30% Overlap	-0.5244	0.5244
C	Diff. Exp. Non-Linear 10% Overlap	± 2.5632	0
D	Diff. Exp. Non-Linear 30% Overlap	± 1.0488	0
E	No Diff. Exp. 100% Overlap	0	0

two). Feature sets in each class are modeled as simple Gaussian distributions. Likewise, the synthetic datasets with two-dimensional feature sets are two-dimensional Gaussian distributions with 10% or 30% overlap (**Figure 16**, top row). Non-linearly separable feature sets are modeled as combinations of Gaussian distributions such that the classes cannot be optimally separated using a linear hyperplane (**Figure 15**, rows three and four, **Figure 16**, bottom row). By varying the noise level of the linear and non-linear datasets, we can assess the robustness of ranking metric optimization to variations in the distribution of differentially expressed feature sets. However, because differentially expressed feature sets in a clinical dataset are seldom similarly distributed, we also use a synthetic dataset of mixed distributions to test the robustness of this method to heterogeneous data.

4.2.7 Microarray Data Analysis and qRT-PCR Validation

We examine four clinical endpoints using various cancer microarray datasets: renal, prostate, and breast cancer. The renal cancer data includes two clinical endpoints and each of the prostate and breast cancer datasets include one endpoint. Furthermore, we use two independent microarray datasets for each endpoint. **Table 7** summarizes the clinical data endpoints.

The renal cancer datasets are derived from two independent studies. The first

Table 5: Two-dimensional Gaussian distribution means for relevant and irrelevant feature sets in the synthetic dataset. Standard deviations of Gaussian distributions are one.

Group Label	Feature Set Group	Class 1 Mean	Class 2 Mean
A	Diff. Exp. Linear 10% Overlap	x: -0.9062	x: 0.9062
		y: -0.9062	y: 0.9062
B	Diff. Exp. Linear 30% Overlap	x: -0.3708	x: 0.3708
		y: -0.3708	y: 0.3708
C	Diff. Exp. Non-Linear 10% Overlap	x: $\mp$ 1.2816	x: $\mp$ 1.2816
		y: $\mp$ 1.2816	y: $\pm$ 1.2816
D	Diff. Exp. Non-Linear 30% Overlap	x: $\mp$ 0.5244	x: $\mp$ 0.5244
		y: $\mp$ 0.5244	y: $\pm$ 0.5244
E	No Diff. Exp. 100% Overlap	0	0

Table 6: Distribution of feature sets for synthetic data.

Feature Set Number	Linear 10%	Linear 30%	Non-linear 10%	Non-linear 30%	Mixed 10%	Mixed 30%
1-50	A	B	C	D	A	B
50-100	A	B	C	D	C	D
101-3000	E	E	E	E	E	E

Table 7: Clinical microarray datasets for knowledge-guided biomarker identification. For each end point, we use two microarray datasets to examine the effect of knowledge-guided feature selection. Each dataset varies in terms of clinical endpoint as well as sample size. Knowledge genes used to assess biomarker detection efficiency are identified from literature as well as from qRT-PCR experiments.

Dataset Code	Endpoint Code	# Knowledge Genes/Probesets	Endpoint Description	Dataset 1		Dataset 2	
				# P	# N	# P	# N
Renal	A	21	Clear Cell vs Onco./Chromo.	13	7	32	18
	B	12	Clear Cell vs Papillary	13	5	32	11
Prostate	C	6	Tumor vs Norm. Adj. Tissue	52	50	61	63
Breast	D	8	pCR vs RD	21	60	12	37

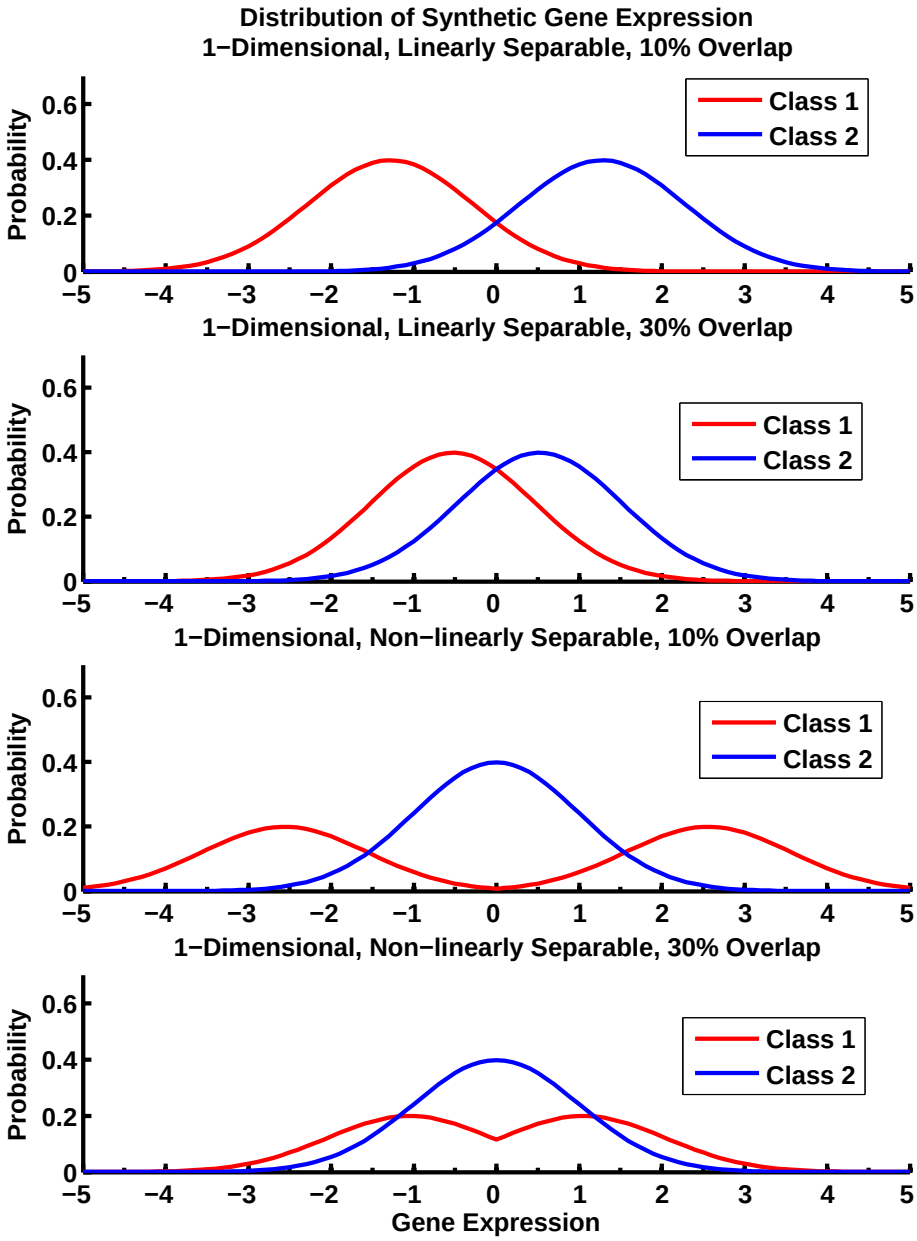


Figure 15: One-Dimensional Synthetic Gene Expression Data Distributions. Each class distribution is Gaussian with variance of one. Linearly separable with 10% overlap between classes (top row). Linearly separable with 30% overlap between classes (second row). Non-linearly separable with 10% overlap between classes (third row). Non-linearly separable with 30% overlap between classes (bottom row).

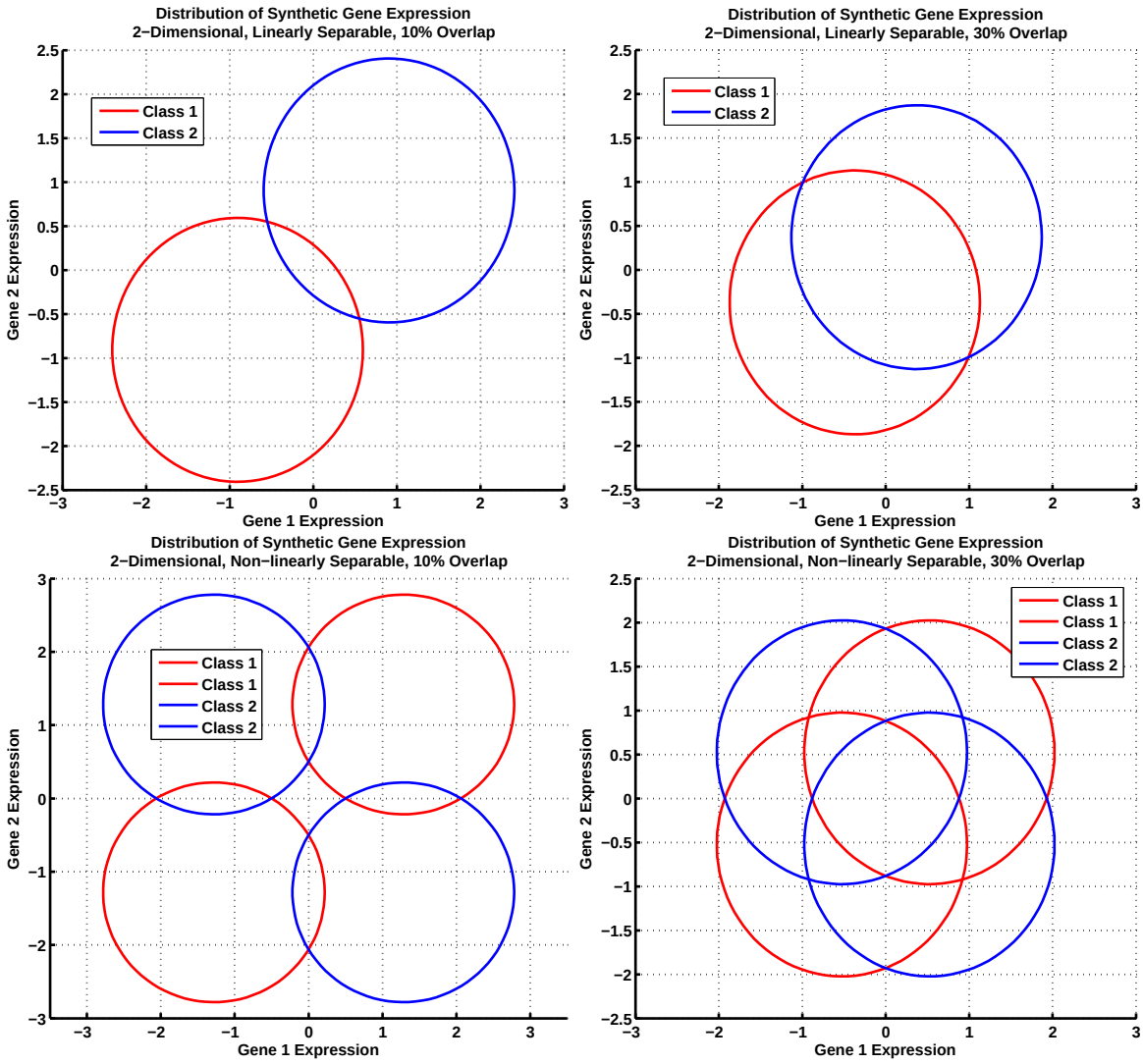


Figure 16: Two-Dimensional Synthetic Gene Expression Data Distributions. Circles represent Gaussian distributions with variance of one. Linearly separable with 10% overlap between classes (top left). Linearly separable with 30% overlap between classes (top right). Non-linearly separable with 10% overlap between classes (bottom left). Non-linearly separable with 30% overlap between classes.

dataset, from a study by Schuetz *et al.*, uses Affymetrix microarrays (HG-Focus, 8793 probesets) to profile samples from several subtypes of renal tumors: 13 clear cell (CC) renal cell carcinoma (RCC), 4 chromophobe (CHR) RCC, and 3 oncocytoma (ONC, benign), and 5 papillary (PAP) [124]. The second dataset, from a study by Jones *et al.*, uses a different model of Affymetrix microarrays (HG-U133A, 22283 probesets reduced to 8793 that are common to HG-Focus) to examine similar renal tumor subtypes with 32 CC, 6 CHR, 12 ONC samples, and 11 PAP [67]. We are interested in biomarkers that differentiate the CC class from the combined group of ONC and CHR (endpoint A) as well as biomarkers that differentiate the CC class from the PAP class (endpoint B).

The prostate cancer datasets are derived from independent studies identifying biomarkers that can differentiate tumor tissue from normal prostate tissue adjacent to tumors [133, 19]. Both of these studies use Affymetrix HG-U95Av2 microarrays and have a relatively large number of samples and probesets compared to the renal cancer endpoints. The breast cancer datasets are derived from a study that focuses on identifying biomarkers that can identify patients most likely to respond to chemotherapy [53]. These datasets were assayed on Affymetrix HG-U133A microarrays and contain 22283 probesets.

Using literature, we identify genes that have been validated (via qRT-PCR or IHC) as differentially expressed between the disease/patient groups for each of the clinical endpoints. We then validate an additional 94 genes using qRT-PCR (using RNA from 34 CC and 18 CHR tissue samples) for the renal cancer endpoints. These 94 genes were selected by a renal cancer pathologist based on his knowledge and previous research. Only some of the 94 genes assayed with qRT-PCR are differentially expressed in the two clinical scenarios as assessed by a linear SVM with classification error estimated using 0.632 bootstrap. Genes measured with qRT-PCR are categorized as differentially expressed if the estimated classification error is less than 10%

(or 20% in the case of the CC vs PAP comparison). Using the set of knowledge from both literature and qRT-PCR validation, we examine the efficiency of detecting these biomarkers for each clinical endpoint by optimizing the ranking metric under various conditions, as illustrated in **Figure 14**.

4.3 Results and Discussion

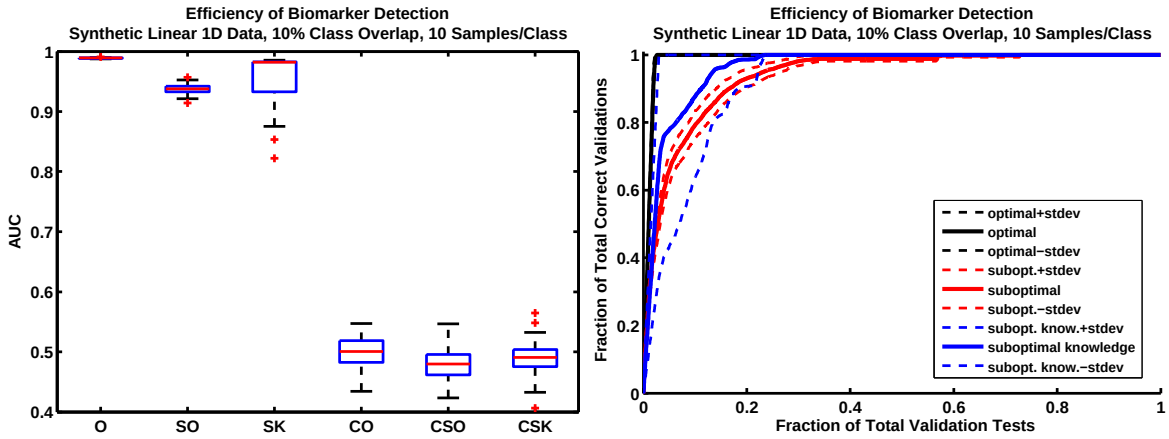
4.3.1 Synthetic Data Simulations

In most cases, when using the maximum likelihood approach, optimizing the ranking metric improves the efficiency of detecting differentially expressed feature sets in synthetic data (**Figure 17**, **Figure 18**, **Figure 19**, **Figure 20**, **Figure 21**, and **Figure 22**). In each of these figures, the solid black line is the detection-efficiency curve for the optimally selected ranking metric. This line is an average of 100 curves, each of which is generated by a bootstrap iteration (described in the previous section). The dashed line indicates the standard deviation of the 100 iterations. The box plots represent the area under the curve (AUC) and summarize the performance of feature set detection ('O' = optimal, 'SO' = sub-optimal, and 'SK' = sub-optimal initial knowledge). We also examine the effect of increasing noise (results of 10% noise are in first column of each figure and results for 30% are in the second column). In all cases, the optimal algorithm performs very well in low noise scenarios. Performance decreases when we increase data noise to 30%, most notably in the non-linear one- and two- dimensional cases (**Figure 18(b)**, **Figure 21(b)**). Obviously, the larger noise level increases the difficulty of detecting differentially expressed feature sets in general, especially in such small datasets. Thus, increasing the synthetic data sample size from 10 samples per class to 20 samples per class improves detection efficiency (**Figure 17(c)**, **Figure 18(c)**, **Figure 19(c)**, **Figure 20(c)**, **Figure 21(c)**, and **Figure 22(c)**).

As expected, choosing a sub-optimal ranking metric reduces the efficiency of detecting differentially expressed feature sets, as indicated by a decrease in the AUCs (red line). Using sub-optimal initial knowledge also decreases the efficiency of detecting these feature sets, but it also significantly increases the variance of the detection curves (blue line). This result is consistent among all synthetic datasets. In some cases, the sub-optimal initial knowledge simulation initially performs worse than the sub-optimal simulation, but improves as the simulation progresses. This may be explained by the iterative addition of truly differentially expressed feature sets to the knowledge set. Thus, even if we begin with a poor set of prior knowledge, if we perform enough validations to progressively improve the knowledge set, then validation efficiency will increase. When ranking feature sets using a sub-optimal ranking metric or sub-optimal initial knowledge, we can expect an increase in the false discovery rate. Translating to a clinical scenario, because validation requires time and resources, we should expect that an increase in the false discovery rate would also reduce the efficiency of biomarker validation.

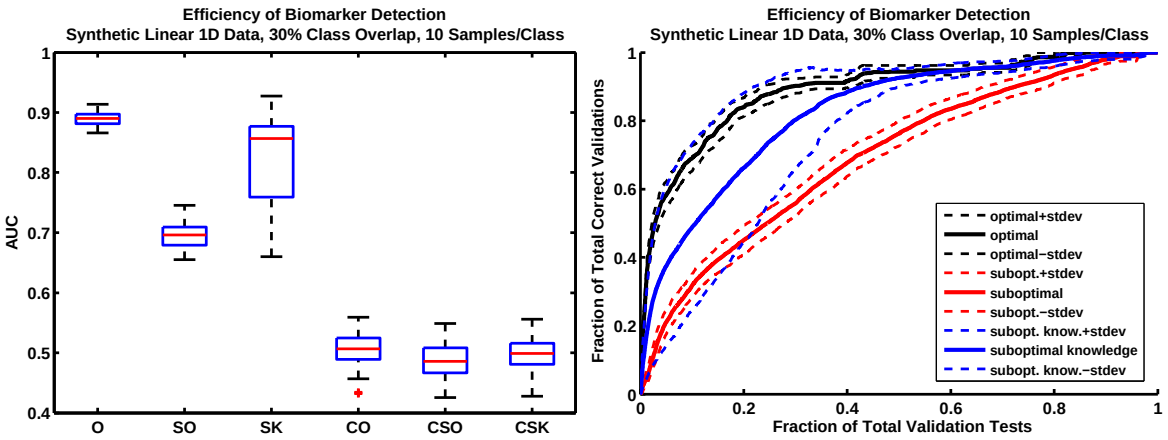
Finally, optimizing the ranking metric using a mixture of linearly and non-linearly separable feature sets does not decrease the effect of optimization (**Figure 19**, **Figure 22**). In these cases, because the initial knowledge set contains both linearly and non-linearly separable feature sets, we can assume that the optimization process identifies a metric that can adequately identify both types of feature sets.

The box plots of each figure also include control cases in which we use randomly selected knowledge from the feature sets that are not differentially expressed. These box plots are labeled with 'CO' = control optimal, 'CSO' = control sub-optimal, and 'CSK' = control sub-optimal initial knowledge. As expected the AUCs of each of these controls are close to 0.5.

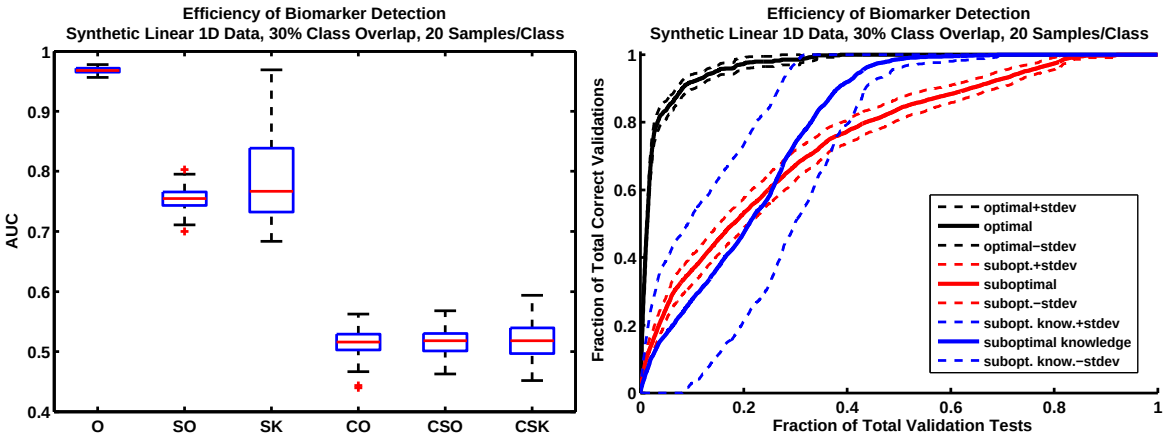


(a) Synthetic one-dimensional linearly separable data simulations with 10% class overlap and 10 samples per class.

Figure 17: The optimally selected ranking metric (black line) is more efficient in detecting differentially expressed feature sets compared to the sub-optimal metrics (blue and red lines). This is true for the low noise case (**17(a)**), high noise case (**17(b)**, next page), and high noise case with larger sample size (**17(c)**, next page). The left figures contain box plots of the area under the ROC curve for the three simulations-optimal algorithm (O), sub-optimal algorithm (SO), and sub-optimal initial knowledge (SK)-as well as three control cases in which knowledge was randomly selected (CO, CSO, and CSK). The control tests show that choice of knowledge is important. Noise generally increases the difficulty of identifying differentially expressed genes (**17(b)**, next page), even with the introduction of prior knowledge. However, increasing the sample size reduces the false discovery rate, resulting in a higher AUC for the optimal algorithm (**17(c)**, next page).

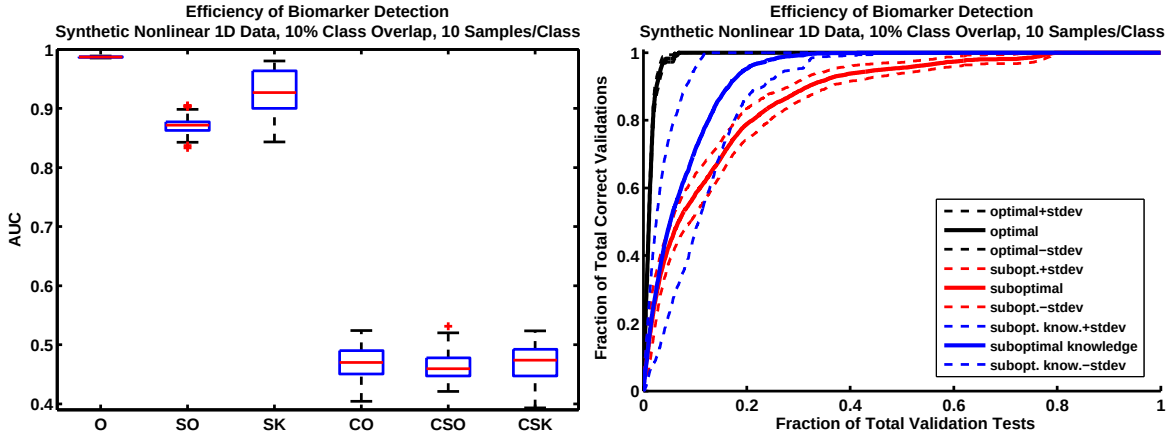


(b) Synthetic one-dimensional linearly separable data simulations with 30% class overlap and 10 samples per class.



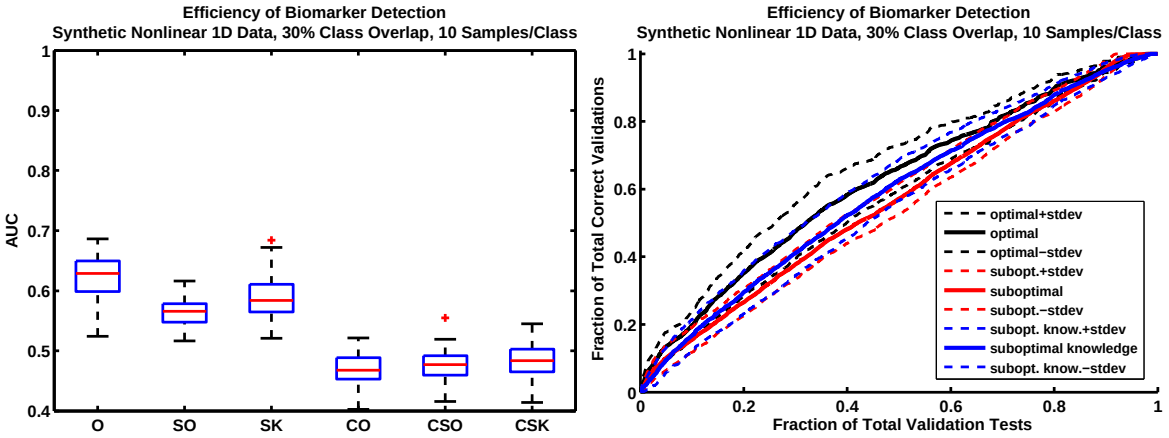
(c) Synthetic one-dimensional linearly separable data simulations with 30% class overlap and 20 samples per class.

Figure 17 parts (b) and (c). Figure part (a) and full caption on the previous page.

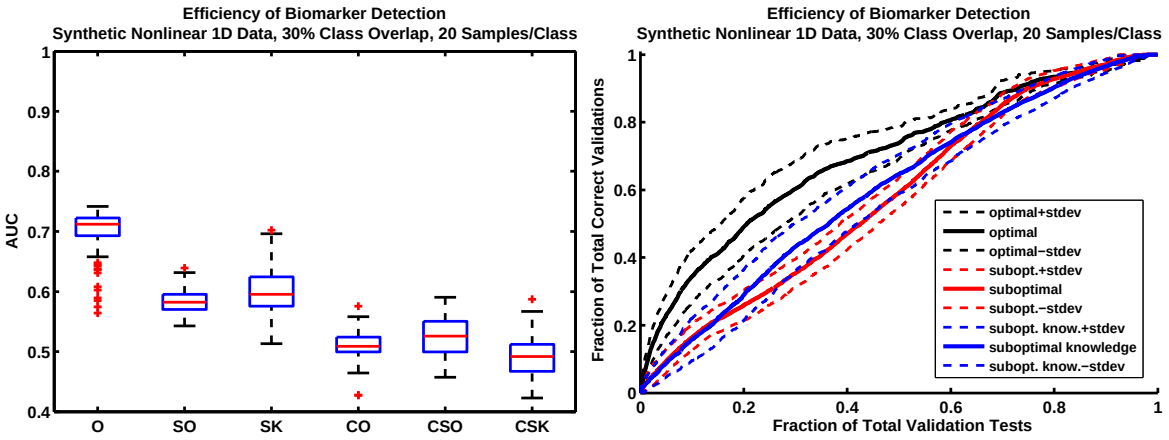


(a) Synthetic one-dimensional non-linearly separable data simulations with 10% class overlap and 10 samples per class.

Figure 18: Synthetic one-dimensional non-linearly separable data simulations with varying noise levels and sample size. Similar to the linearly separable case, the optimally selected ranking metric (black line) for the non-linearly separable case is more efficient in detecting differentially expressed feature sets compared to the sub-optimal metrics (blue and red lines). However, as noise increases, the false discovery rate also increases significantly (**18(b)**, next page). Again, the addition of samples reduces the false discovery rate (**18(c)**, next page). The left figures contain box plots of the area under the ROC curve for the three simulations-optimal algorithm (O), sub-optimal algorithm (SO), and sub-optimal initial knowledge (SK)-as well as three control cases in which knowledge was randomly selected (CO, CSO, and CSK). The control tests show that choice of knowledge is important.

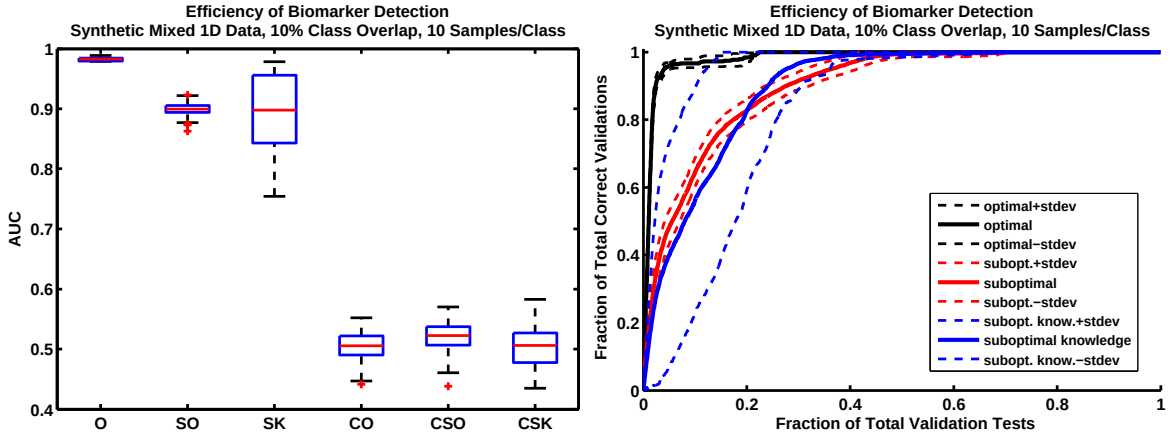


(b) Synthetic one-dimensional non-linearly separable data simulations with 30% class overlap and 10 samples per class.



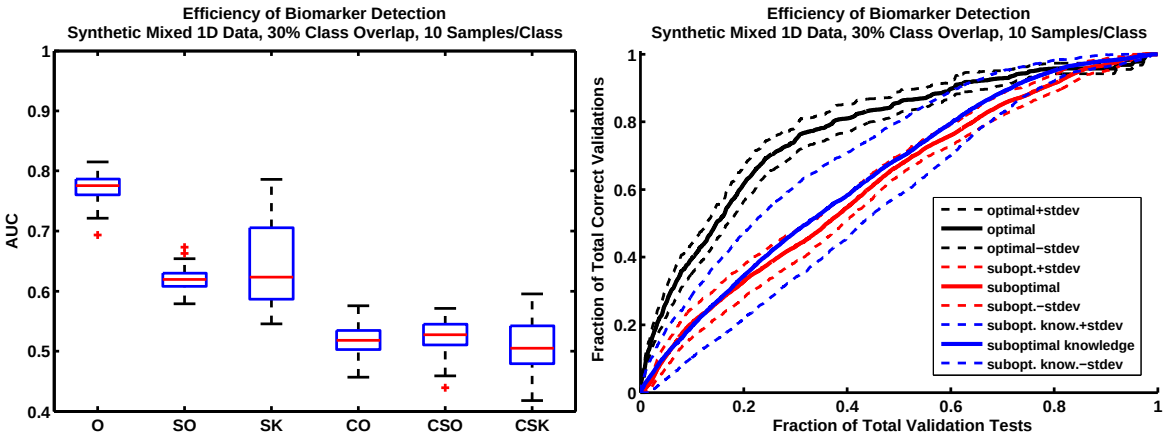
(c) Synthetic one-dimensional non-linearly separable data simulations with 30% class overlap and 20 samples per class.

Figure 18 parts (b) and (c). Figure part (a) and full caption on the previous page.

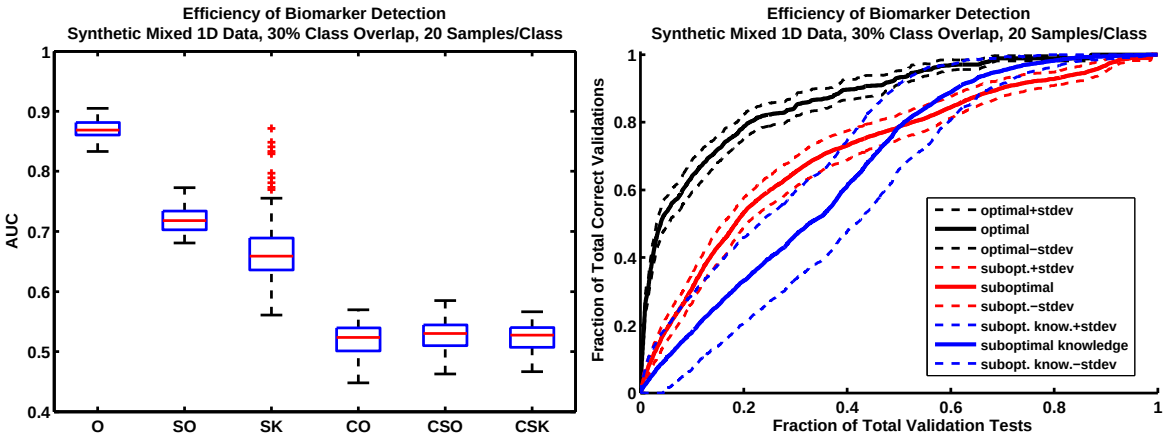


(a) Synthetic one-dimensional mixed data simulations with 10% class overlap and 10 samples per class.

Figure 19: Synthetic one-dimensional mixed data simulations with varying noise levels and sample size. Mixed data consists of both linearly and non-linearly separable biomarkers. Similar to the linearly and non-linearly separable cases, the optimally selected ranking metric (black line) for the mixed case is more efficient in detecting differentially expressed feature sets compared to the sub-optimal metrics (blue and red lines). Once again, increases in noise increases the false discovery rate (**19(b)**, next page), but addition of samples decreases the false discovery rate (**19(c)**, next page). The top row contains box plots of the area under the ROC curve for the three simulations-optimal algorithm (O), sub-optimal algorithm (SO), and sub-optimal initial knowledge (SK)-as well as three control cases in which knowledge was randomly selected (CO, CSO, and CSK). The control tests show that choice of knowledge is important. This simulation shows that, even in the presence of a mixture of biomarker distributions, a diverse population of feature selection methods will still yield a method that performs well.

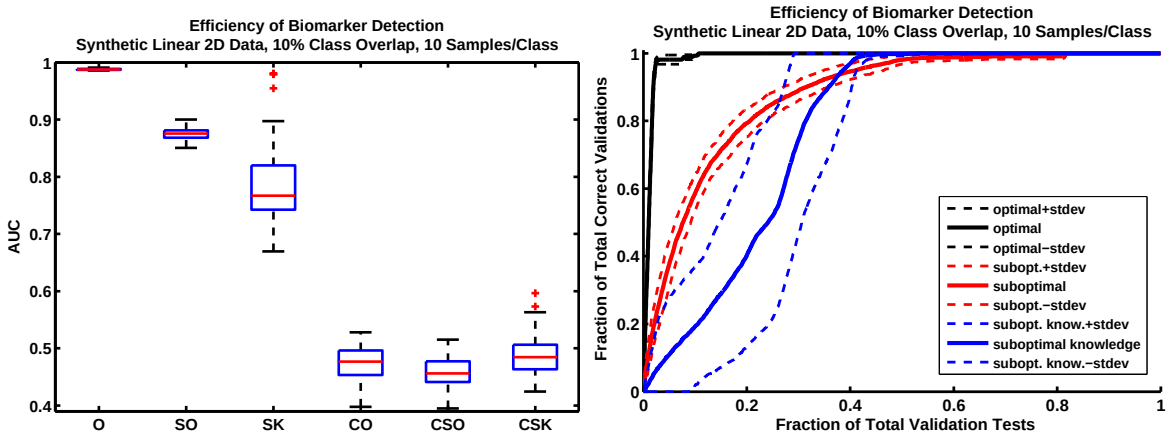


(b) Synthetic one-dimensional mixed data simulations with 30% class overlap and 10 samples per class.



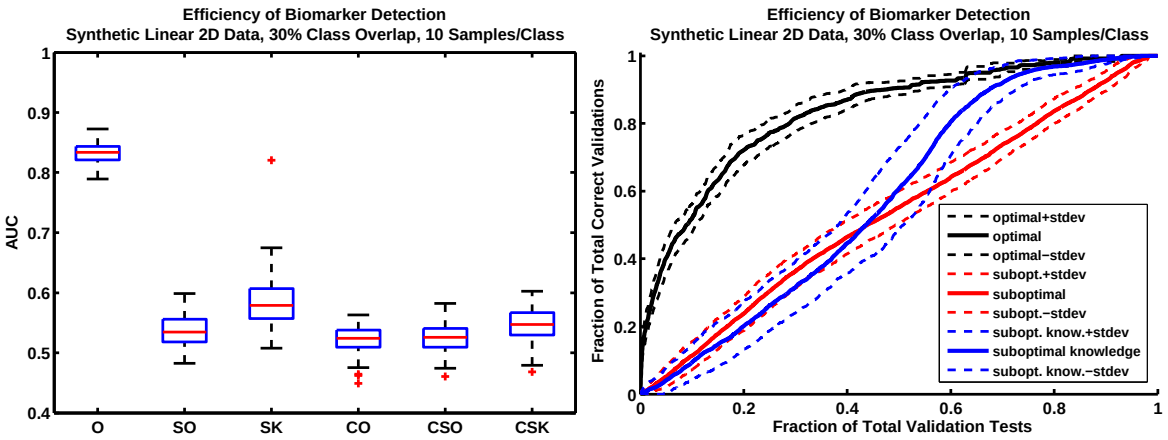
(c) Synthetic one-dimensional mixed data simulations with 30% class overlap and 20 samples per class.

Figure 19 parts (b) and (c). Figure part (a) and full caption on the previous page.

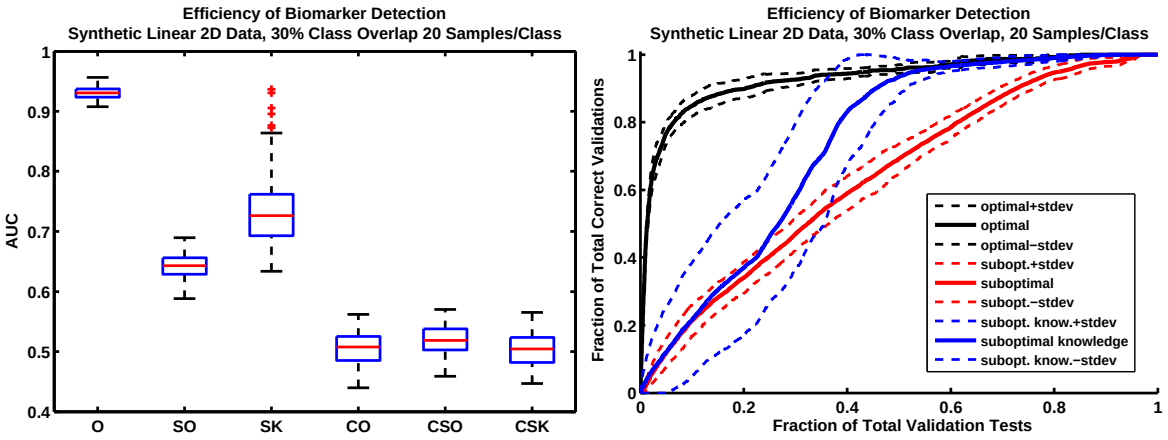


(a) Synthetic two-dimensional linearly separable data simulations with 10% class overlap and 10 samples per class.

Figure 20: Synthetic two-dimensional linearly separable data simulations with varying noise levels and sample size. The optimally selected ranking metric (black line) is more efficient in detecting differentially expressed feature sets compared to the sub-optimal metrics (blue and red lines). This is true for the low noise case (**20(a)**), high noise case (**20(b)**, next page), and high noise case with larger sample size (**20(c)**, next page). The left figures contain box plots of the area under the ROC curve for the three simulations-optimal algorithm (O), sub-optimal algorithm (SO), and sub-optimal initial knowledge (SK)-as well as three control cases in which knowledge was randomly selected (CO, CSO, and CSK). The control tests show that choice of knowledge is important. Noise generally increases the difficulty of identifying differentially expressed genes (middle column), even with the introduction of prior knowledge. However, increasing the sample size reduces the false discovery rate, resulting in a higher AUC for the optimal algorithm (**20(c)**, next page).

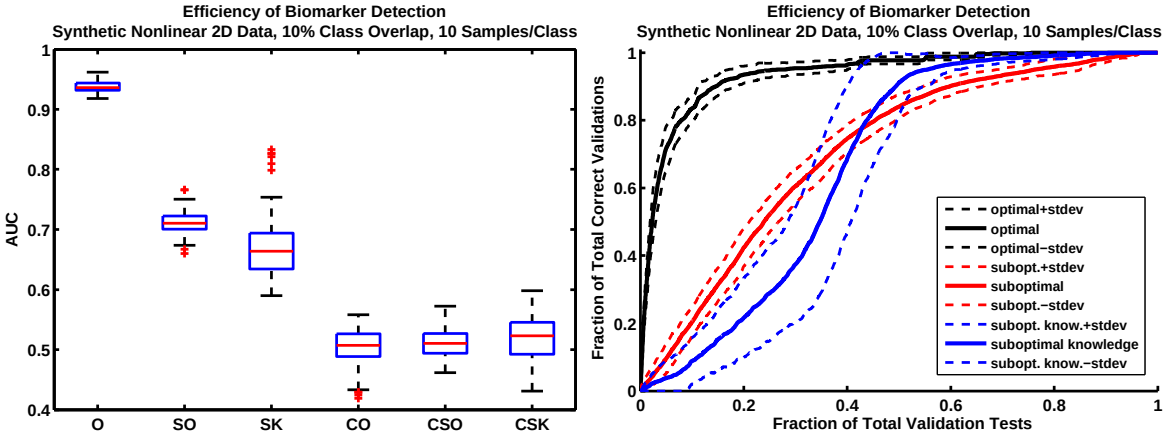


(b) Synthetic two-dimensional linearly separable data simulations with 30% class overlap and 10 samples per class.



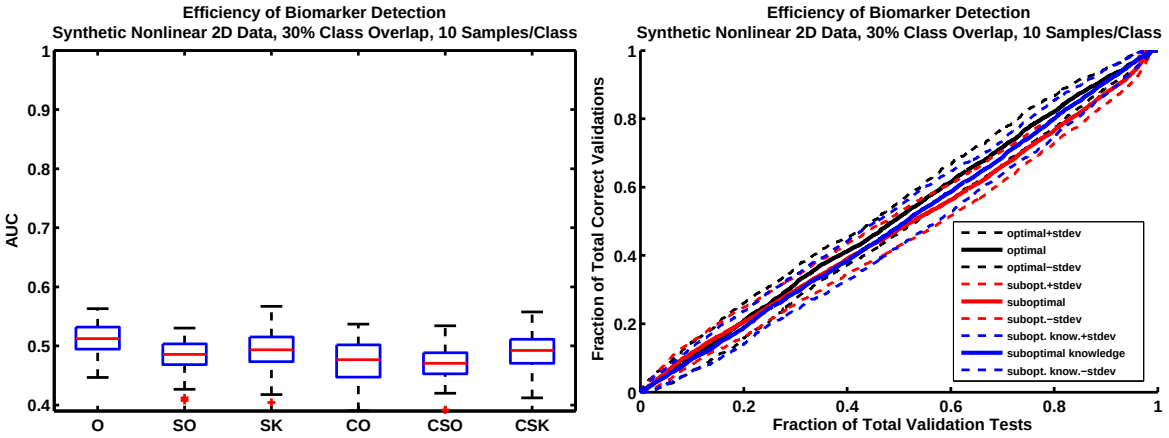
(c) Synthetic two-dimensional linearly separable data simulations with 30% class overlap and 20 samples per class.

Figure 20 parts (b) and (c). Figure part (a) and full caption on the previous page.

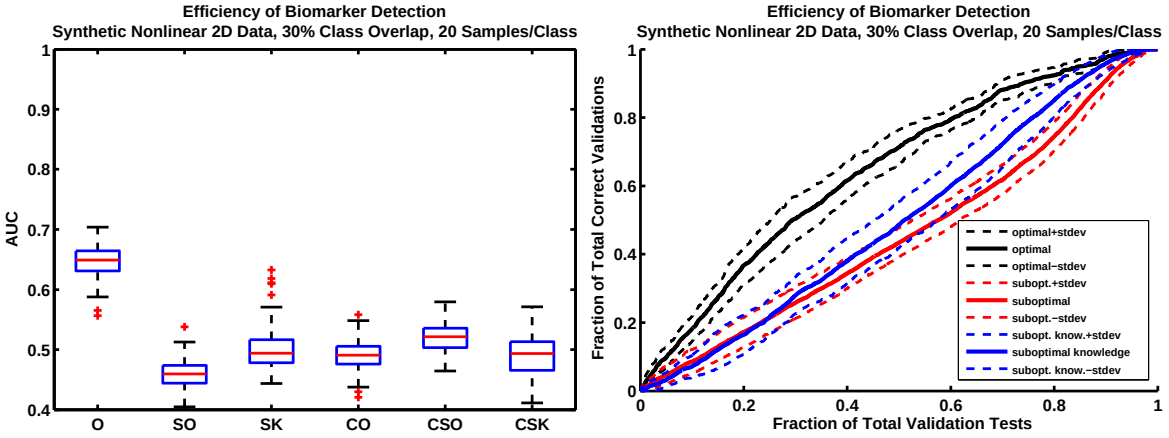


(a) Synthetic two-dimensional non-linearly separable data simulations with 10% class overlap and 10 samples per class.

Figure 21: Synthetic two-dimensional non-linearly separable data simulations with varying noise levels and sample size. Similar to the linearly separable case, the optimally selected ranking metric (black line) for the non-linearly separable case is more efficient in detecting differentially expressed feature sets compared to the sub-optimal metrics (blue and red lines). However, as noise increases, the false discovery rate also increases significantly (**21(b)**, next page). Again, the addition of samples reduces the false discovery rate (**21(c)**, next page). The top row contains box plots of the area under the ROC curve for the three simulations-optimal algorithm (O), sub-optimal algorithm (SO), and sub-optimal initial knowledge (SK)-as well as three control cases in which knowledge was randomly selected (CO, CSO, and CSK). The control tests show that choice of knowledge is important.

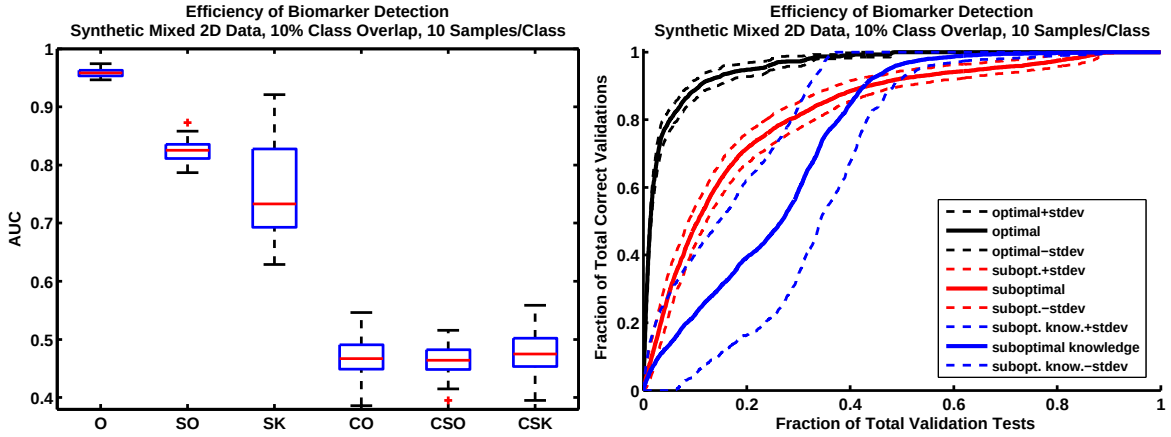


(b) Synthetic two-dimensional non-linearly separable data simulations with 30% class overlap and 10 samples per class.



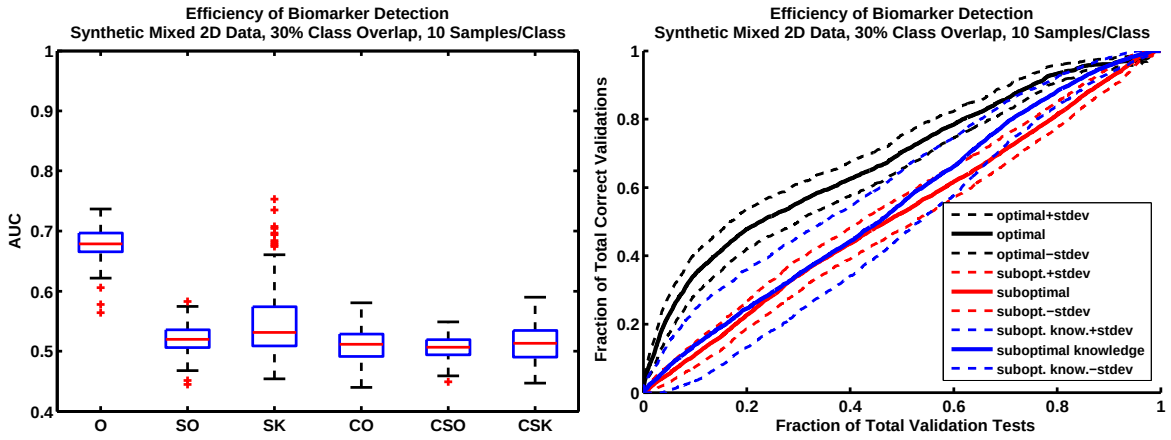
(c) Synthetic two-dimensional non-linearly separable data simulations with 30% class overlap and 20 samples per class.

Figure 21 parts (b) and (c). Figure part (a) and full caption on the previous page.

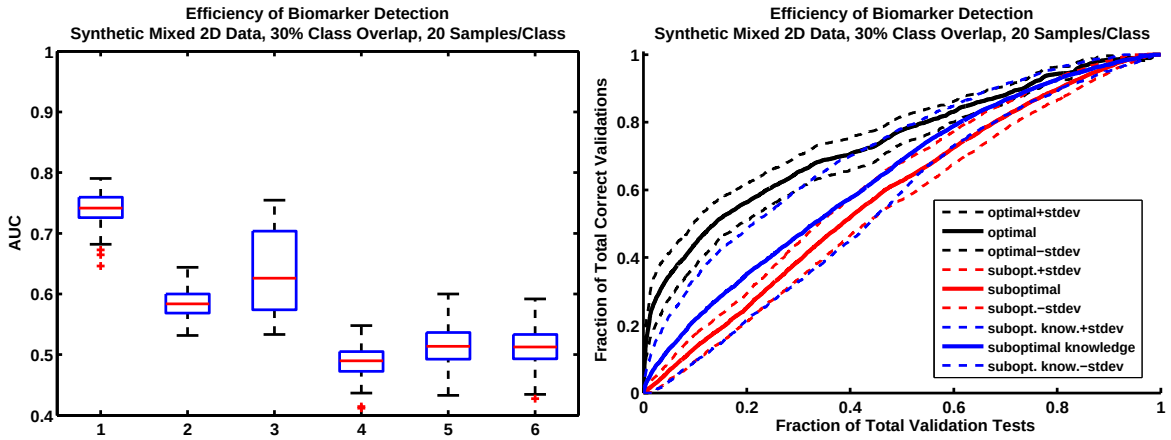


(a) Synthetic two-dimensional mixed data simulations with 10% class overlap and 10 samples per class.

Figure 22: Synthetic two-dimensional mixed data simulations with varying noise levels and sample size. Mixed data consists of both linearly and non-linearly separable biomarkers. Similar to the linearly and non-linearly separable cases, the optimally selected ranking metric (black line) for the mixed case is more efficient in detecting differentially expressed feature sets compared to the sub-optimal metrics (blue and red lines). Once again, increases in noise increases the false discovery rate (**22(b)**, next page), but additional samples decreases the false discovery rate (**22(c)**, next page). The top row contains box plots of the area under the ROC curve for the three simulations-optimal algorithm (O), sub-optimal algorithm (SO), and sub-optimal initial knowledge (SK)-as well as three control cases in which knowledge was randomly selected (CO, CSO, and CSK). The control tests show that choice of knowledge is important. This simulation shows that, even in the presence of a mixture of biomarker distributions, a diverse population of feature selection methods will still yield a method that performs well.



(b) Synthetic two-dimensional mixed data simulations with 30% class overlap and 10 samples per class.



(c) Synthetic two-dimensional mixed data simulations with 30% class overlap and 20 samples per class.

Figure 22 parts (b) and (c). Figure part (a) and full caption on the previous page.

4.3.2 Clinical Data Analysis

As described in the methods, we identify several biomarkers from literature that are differentially expressed for each of the four clinical endpoints. For endpoints A and B, the renal cancer endpoints, we identified additional biomarkers using qRT-PCR validation. **Table 8** lists the renal cancer biomarkers identified from literature for endpoints A1 and A2. qRT-PCR validated biomarkers for endpoints A and B are listed in **Table 9** and **Table 10**, respectively. We filter the qRT-PCR validated biomarkers such that their estimated classification errors are less than 10%. Biomarker for prostate and breast cancer are more difficult to identify, mainly due to the heterogeneity of these diseases. The MYC and FASN (fatty acid synthase) genes are differentially expressed in normal and prostate cancer tissue as assessed by immunohistochemistry [110]. Schlomm *et al.* also identified and validated MYC as well as FOLH1 using qRT-PCR [122]. Hepsin (HPN), AMACR, and SARDH have also been validated using various methods [136, 85, 121]. These biomarkers are listed in **Table 11**. It is important that the biomarkers used as knowledge to guide feature selection be specific to the disease in question. For example, biomarkers that are differentially expressed between metastatic prostate cancer and benign prostate cancer may not be appropriate for distinguishing prostate tumors from normal tissue. Likewise, we identify the MAPT and KI67 biomarkers from literature that are differentially expressed between breast cancer treatment success and failure (**Table 12**). Using immunohistochemistry, Rouzier *et al.* identified and validated the microtubule-associated protein tau (MAPT) as a biomarker for paclitaxel (chemotherapy) sensitivity in breast cancer [120]. The Ki-67 biomarker was shown to have a strong correlation with pathological response to chemotherapy in breast cancer patients [15]. Because of the variety of chemotherapy regimens for breast cancer, knowledge biomarkers should be carefully chosen to closely reflect the dataset of interest. Here, we examine the pathologic

response of breast cancer patients to a common treatment regimen, T/FAC. Biomarkers that can predict success or failure to this particular treatment regimen may not accurately predict success or failure for a different treatment regimen.

Combining all knowledge from both literature and qRT-PCR validation, we examine the effect of optimizing the feature ranking metric using the method illustrated in **Figure 14**. For the CC vs ONC/CHR subtype comparison, endpoints A1 and A2, box plots of the 100 iterations for each of the three tests indicate that knowledge guided feature selection method using both MLE and MAP (black lines) outperforms the standard significance analysis of microarrays (SAM, green line) filter method (**Figure 23**). The Bayesian maximum *a posteriori* method (solid line) slightly outperforms the MLE method (dashed line) when the initial knowledge is of good quality. In the scenario where the initial knowledge is randomly selected from the set of all genes, the MAP method significantly outperforms the MLE method (red lines). The box plots (**Figure 23**, left column) represent the median and quartiles of the AUC values for each of the 100 iterations. The results are similar for the renal cancer endpoints B1 and B2, although the improvement from using the Bayesian MAP approach for endpoint B2 is not significant (**Figure 24**). As expected, the control tests, in which we are detecting randomly selected genes using randomly selected initial knowledge, result in AUCs of approximately 0.5. This indicates that none of the feature selection methods favors uninformative genes better than random chance.

For the prostate cancer endpoints, C1 and C2, the knowledge-guide biomarker identification algorithm appears to be less efficient compared to SAM (**Figure 25**). This result may seem unintuitive since the SAM method is included in the population of feature selection methods that the knowledge-guided algorithm considers. If we were to consider all knowledge genes identified for the prostate cancer endpoints in **Table 11**, the best method would likely be SAM. However, due to the bootstrapping that occurs in the simulation, only a subset of the knowledge genes are considered

for selecting the best algorithm. Therefore, the resulting decrease in detection performance is likely due to the fact that the knowledge set only contains a relatively small number of genes and that these genes may not be representative of the best biomarkers for this particular clinical endpoint. Nevertheless, we still see relatively good performance of the knowledge-guided biomarker detection algorithm compared to the sub-optimal initial knowledge simulation. Again, there is a slight improvement of the Bayesian MAP method for the sub-optimal initial knowledge simulation, although the improvement is not statistically significant.

Results are similar for the breast cancer endpoints, D1 and D2, (**Figure 26**). In this simulation, we see that the knowledge-guided biomarker detection algorithm performs better than fold change filtering for the D1 endpoint and similarly for the D2 endpoint. Again, we see that the Bayesian MAP method improves detection efficiency, especially for the sub-optimal initial knowledge simulations (red lines).

The high variance of the suboptimal initial knowledge condition indicates that optimization of the ranking metric is sensitive to the initial conditions. Some of the randomly selected initial knowledge may, in fact, be differentially expressed, resulting in good performance. However, these random initial knowledge sets are more likely to be irrelevant. Thus, box plots for this condition illustrate this mixture of knowledge quality. These results stress the importance of the quality of biomarker knowledge.

Table 8: Genes identified from literature as differentially expressed between renal cancer CC and ONC/CHR subtypes.

Gene Symbol	Knowledge Source	Validation Method
CA9	Chen, Clin Cancer Res, 2005	qRT-PCR
CLCNKB	Chen, Clin Cancer Res, 2005	qRT-PCR
DEFB1	Schuetz, J Mol Diagn, 2005	qRT-PCR, IHC
LRP2	Schuetz, J Mol Diagn, 2005	qRT-PCR, IHC
PVALB	Chen, Clin Cancer Res, 2005	qRT-PCR

Table 9: Genes validated with qRT-PCR for the renal cancer CC vs ONC/CHR subtype comparison. These genes have estimated classification errors of less than 10% as assessed by a linear SVM classifier using 0.632 bootstrap estimation.

Gene Symbol	Validation Method	Estimated Error
STC1	qRT-PCR	2.43E-05
SLC25A4	qRT-PCR	0.00186696
CFTR	qRT-PCR	0.00279081
PDHA1	qRT-PCR	0.0133316
PFKM	qRT-PCR	0.0279739
NNMT	qRT-PCR	0.0289622
CP	qRT-PCR	0.0300157
CFB	qRT-PCR	0.0387219
COX5A	qRT-PCR	0.0394058
BAG1	qRT-PCR	0.0548365
LY6E	qRT-PCR	0.0596081
CD99	qRT-PCR	0.0600892
AKAP12	qRT-PCR	0.0624445
ACAT1	qRT-PCR	0.0687972
SPTBN2	qRT-PCR	0.077287
GOT1	qRT-PCR	0.0784855

Table 10: Genes validated with qRT-PCR for the renal cancer CC vs PAP subtype comparison. These genes have estimated classification errors of less than 20% as assessed by a linear SVM classifier using 0.632 bootstrap estimation.

Gene Symbol	Validation Method	Estimated Error
STC1	qRT-PCR	0.0345774
NDUFA4L2	qRT-PCR	0.0379203
CA9	qRT-PCR	0.0701198
CP	qRT-PCR	0.0781111
ELF3	qRT-PCR	0.0819628
BST2	qRT-PCR	0.112016
B3GNT4	qRT-PCR	0.138581
GRB7	qRT-PCR	0.168125
BAMBI	qRT-PCR	0.169147
CCL20	qRT-PCR	0.188437
CTSC	qRT-PCR	0.192068
PECAM1	qRT-PCR	0.194247

Table 11: Validated prostate cancer biomarkers identified from literature. These biomarkers are specific for distinguishing prostate tumor samples from normal prostate tissue.

Gene Symbol	Source
SARDH	Sreekumar, Nature, 2009
AMACR	Luo, Cancer Research, 2002
HPN	Sardana, Clin Chem, 2008
MYC	Prowatke, British Journal of Cancer, 2007
FASN	Prowatke, British Journal of Cancer, 2007
FOLH1	Schlomm, Eur Urol, 2008

Table 12: Validated breast cancer biomarkers identified from literature. These biomarkers are specific for distinguishing between chemotherapy treatment outcome. Specifically, for pathologic complete response (pCR) to T/FAC treatment versus residual disease after a predefined period of time.

Gene Symbol	Source
MAPT	Rouzier, PNAS, 2005
KI67	Burcombe, Breast Cancer Res, 2006

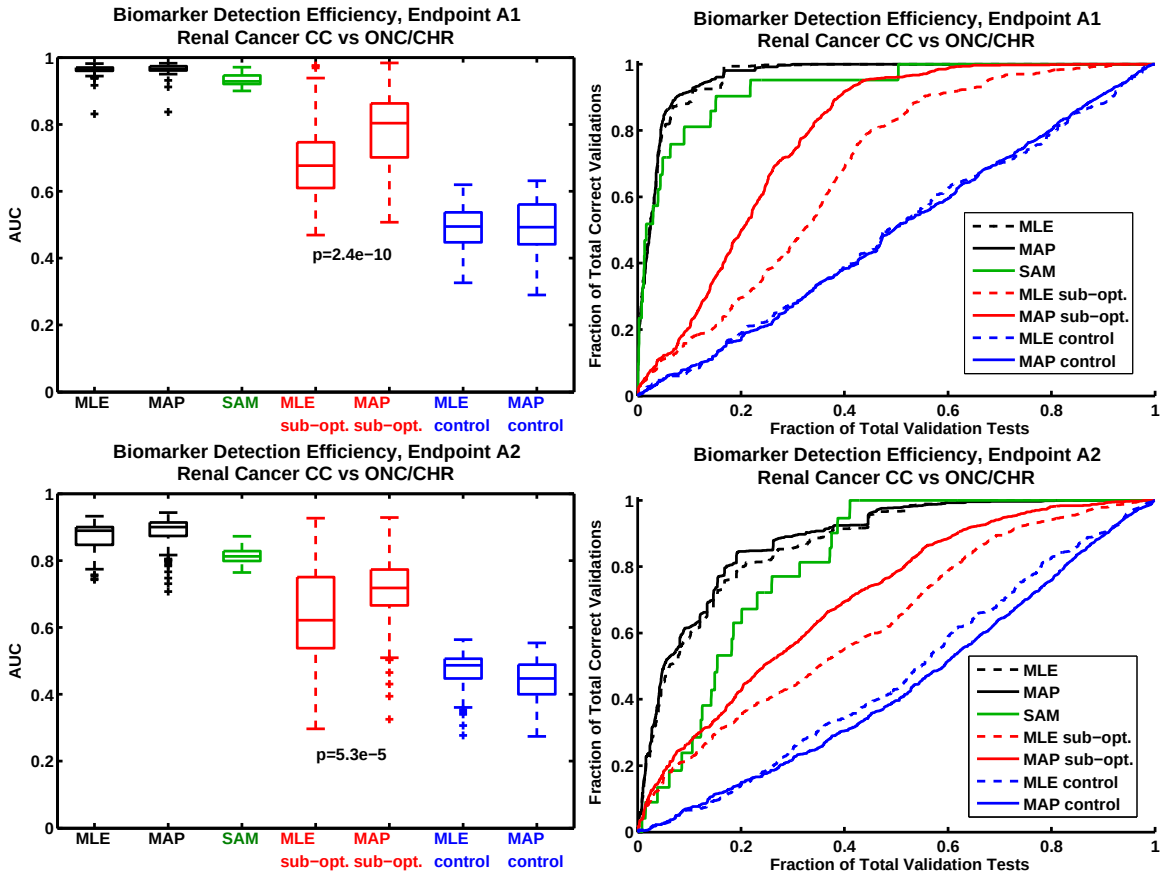


Figure 23: Biomarker detection efficiency assessed for data endpoints A1 and A2, renal cancer CC vs ONC/CHR. The detection efficiency curves (right column) represent average biomarker detection efficiency over 100 bootstrap iterations. Black lines represent optimal ranking metric selection using either the maximum likelihood (MLE, dashed lines) or Bayesian maximum *a posteriori* (MAP, solid lines) methods. Optimal ranking metric selection performs better when the initial knowledge is correct, compared to the case in which the initial knowledge is randomly chosen (red lines). Control tests (blue lines) indicate cases in which all knowledge genes are randomly chosen. As a baseline comparison, we plot the biomarker detection efficiency of SAM, a standard gene filtering method. Overall, SAM performs slightly worse compared to the adaptive knowledge based biomarker detection. The figures in the left column are box plots of area under the curve (AUC) for each of the detection efficiency curves. The Bayesian MAP method performs significantly better than MLE in sub-optimal initial knowledge conditions.

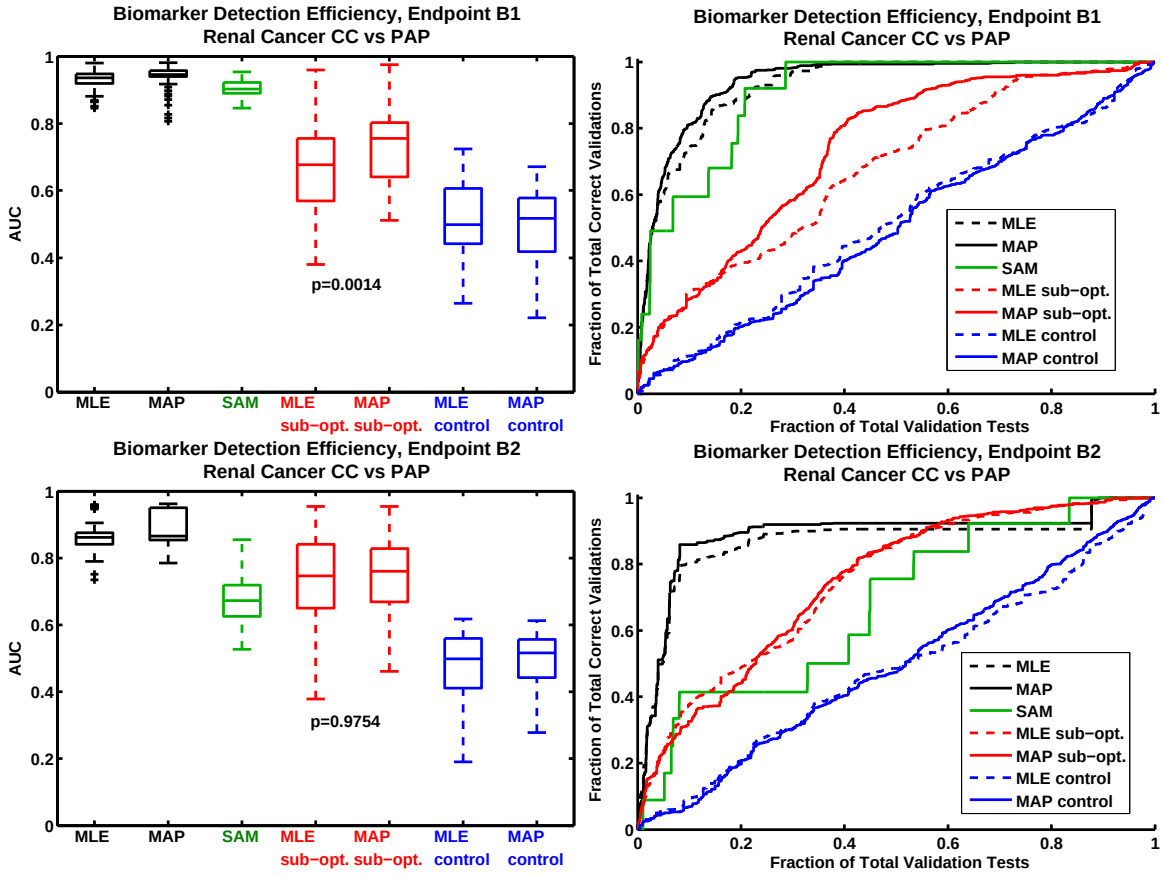


Figure 24: Biomarker detection efficiency assessed for data endpoints B1 and B2, renal cancer CC vs PAP. The detection efficiency curves (right column) represent average biomarker detection efficiency over 100 bootstrap iterations. Black lines represent optimal ranking metric selection using either the maximum likelihood (MLE, dashed lines) or Bayesian maximum *a posteriori* (MAP, solid lines) methods. Optimal ranking metric selection performs better when the initial knowledge is correct, compared to the case in which the initial knowledge is randomly chosen (red lines). Control tests (blue lines) indicate cases in which all knowledge genes are randomly chosen. As a baseline comparison, we plot the biomarker detection efficiency of SAM, a standard gene filtering method. Overall, SAM performs slightly worse compared to the adaptive knowledge based biomarker detection. The figures in the left column are box plots of area under the curve (AUC) for each of the detection efficiency curves. The Bayesian MAP method performs significantly better than MLE in sub-optimal initial knowledge conditions for the B1 endpoint. Bayesian MAP is nominally better than MLE for endpoint B2, but is not statistically significant.

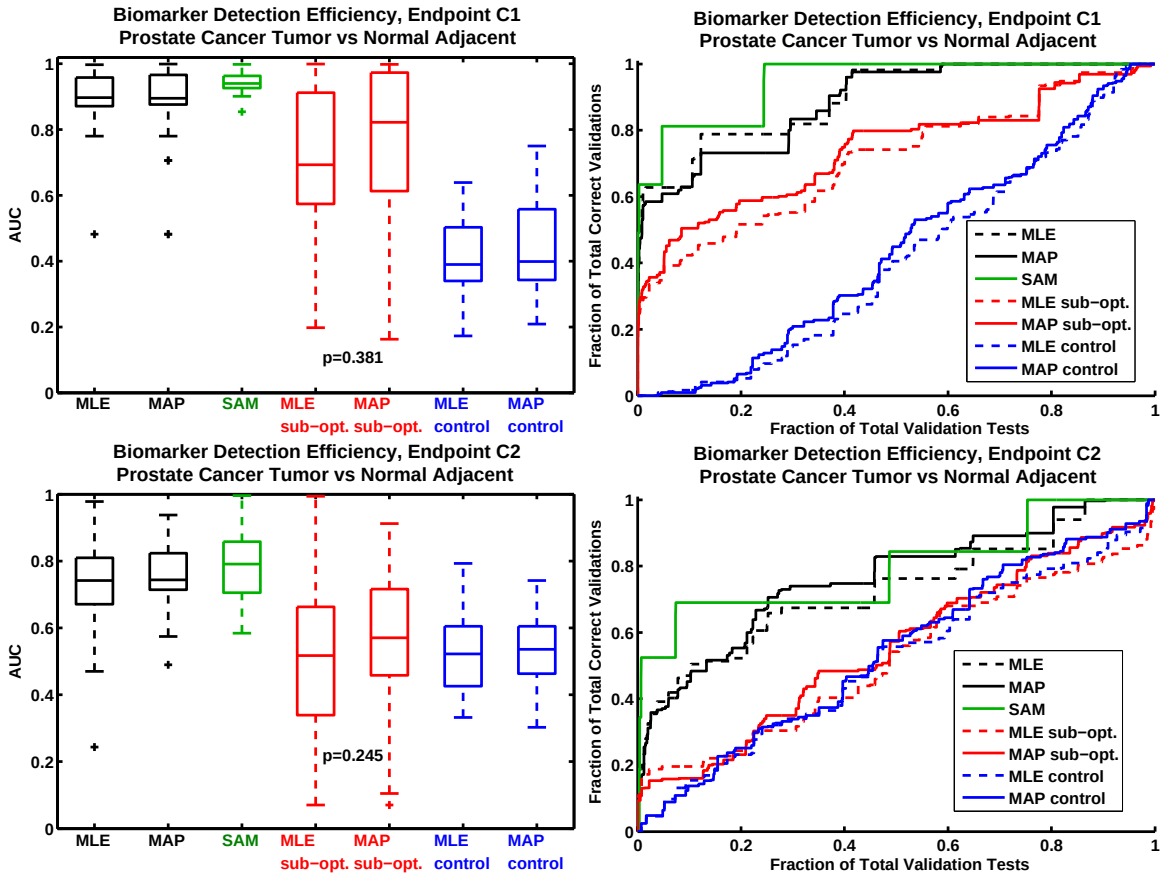


Figure 25: Biomarker detection efficiency assessed for data endpoints C1 and C2, prostate cancer tumor vs normal adjacent tissue. The detection efficiency curves (right column) represent average biomarker detection efficiency over 100 bootstrap iterations. Black lines represent optimal ranking metric selection using either the maximum likelihood (MLE, dashed lines) or Bayesian maximum *a posteriori* (MAP, solid lines) methods. Optimal ranking metric selection performs better when the initial knowledge is correct, compared to the case in which the initial knowledge is randomly chosen (red lines). Control tests (blue lines) indicate cases in which all knowledge genes are randomly chosen. As a baseline comparison, we plot the biomarker detection efficiency of SAM, a standard gene filtering method. Overall, SAM performs slightly better compared to the adaptive knowledge based biomarker detection. The figures in the left column are box plots of area under the curve (AUC) for each of the detection efficiency curves. The Bayesian MAP method performs slightly better than MLE in sub-optimal initial knowledge conditions for both the C1 and C2 endpoints, but the difference is not statistically significant.

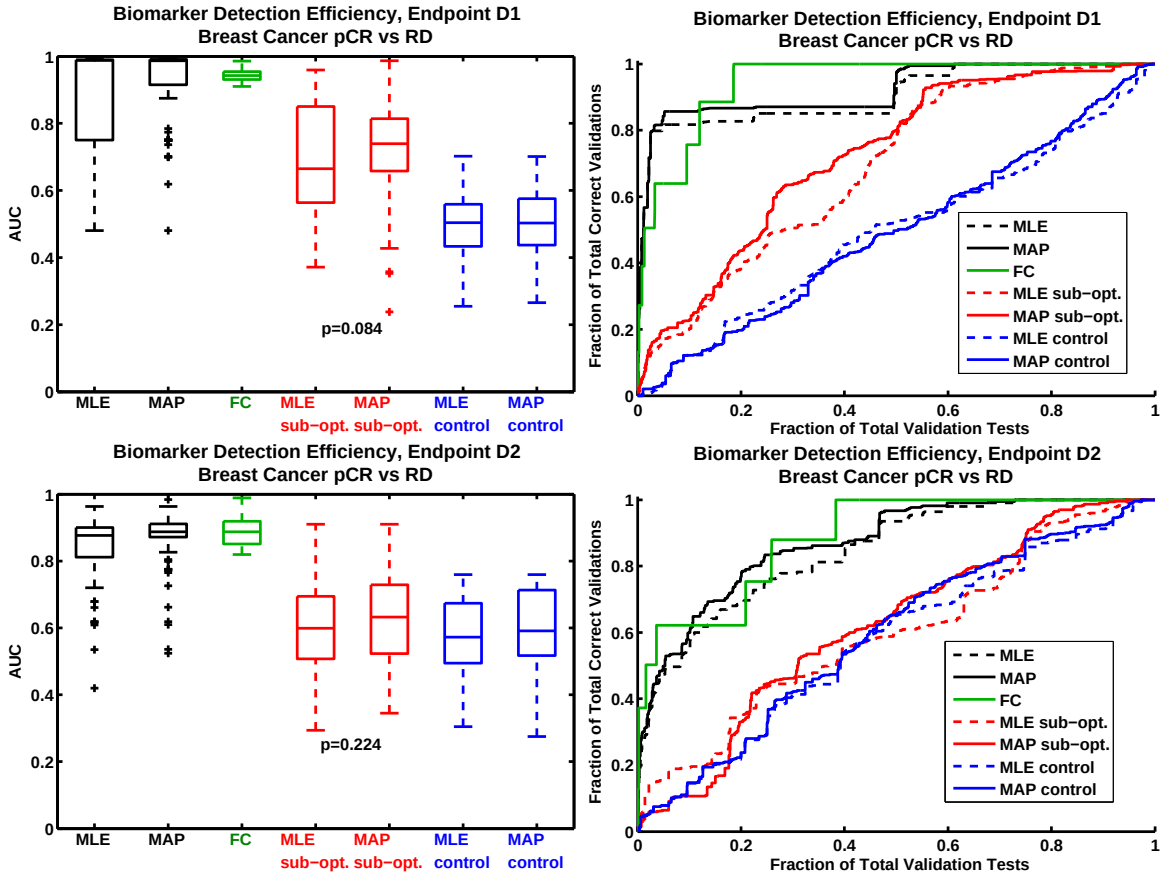


Figure 26: Biomarker detection efficiency assessed for data endpoints D1 and D2, breast cancer pCR vs RD (treatment outcome). The detection efficiency curves (right column) represent average biomarker detection efficiency over 100 bootstrap iterations. Black lines represent optimal ranking metric selection using either the maximum likelihood (MLE, dashed lines) or Bayesian maximum *a posteriori* (MAP, solid lines) methods. Optimal ranking metric selection performs better when the initial knowledge is correct, compared to the case in which the initial knowledge is randomly chosen (red lines). Control tests (blue lines) indicate cases in which all knowledge genes are randomly chosen. As a baseline comparison, we plot the biomarker detection efficiency of the commonly used fold-change ranking method. Overall, fold change performs almost equally compared to the adaptive knowledge based biomarker detection. The figures in the left column are box plots of area under the curve (AUC) for each of the detection efficiency curves. The Bayesian MAP method performs slightly better than MLE in sub-optimal initial knowledge conditions for both the D1 and D2 endpoints, but the difference is not statistically significant for endpoint D2.

4.3.3 Evolution of Ranking Algorithm Probabilities

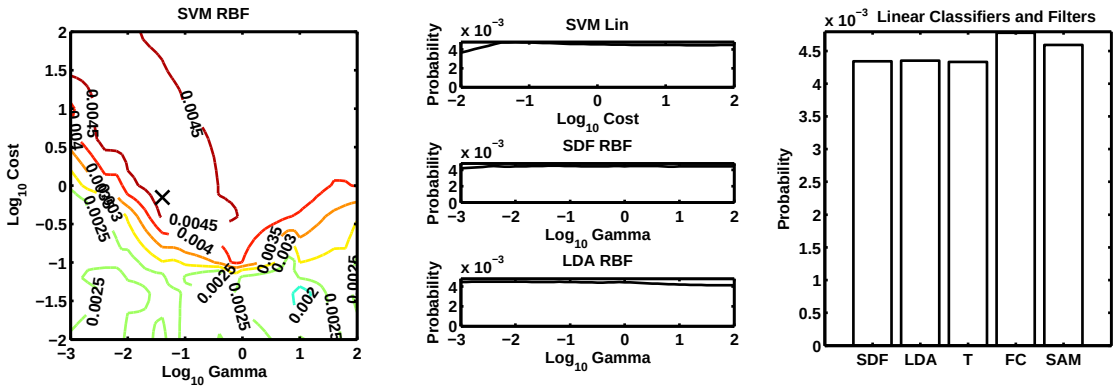
During the biomarker discovery process, as we introduce knowledge, the biological relevance–or probability–of each ranking algorithm changes. The iterative changes reflect our changing knowledge about the clinical problem. Moreover, as the probabilities change, the resulting optimal ranking algorithm should also change. We expect that the probabilities are highly variable when our knowledge is small, but evolves toward the true solution as knowledge increases. Iterative identification of new and validated biomarkers using the maximum likelihood algorithm or the maximum *a posteriori* approach should result in very different changes to the probabilities. Here, we examine the differences between these two methods using the renal cancer dataset that compares CC and ONC/CHR subtypes (endpoints A1 and A2). We use the identified and validated biomarkers for this dataset (**Table 8**, **Table 9**) and the bootstrap method described in the previous section. The probabilities update after discovering each new biomarker. We also use the sub-optimal initial knowledge case to determine which probability update method is more effective.

Figure 27 is the evolution of ranking algorithm probabilities using the maximum likelihood algorithm. From the initial knowledge set (**Figure 27(a)**) to the final iteration after adding approximately 10 new biomarkers as additional knowledge (**Figure 27(c)**) does not significantly change the algorithm probabilities. However, the algorithm with maximum likelihood changes slightly. The contrast in color within the space SVM RBF ranking algorithms indicates the contrast in biological relevance of the methods (**27**, left panels). It is difficult to gauge the benefit of additional knowledge in this scenario due to the very small changes in the algorithm probabilities. As such, we also examine the scenario in which the initial knowledge is of poor quality–i.e., we begin with a knowledge set that includes several randomly selected genes that have not been validated and will likely not be favorably ranked by any ranking metric (**Figure 28**). In contrast to the scenario with correct initial knowledge, the algorithm

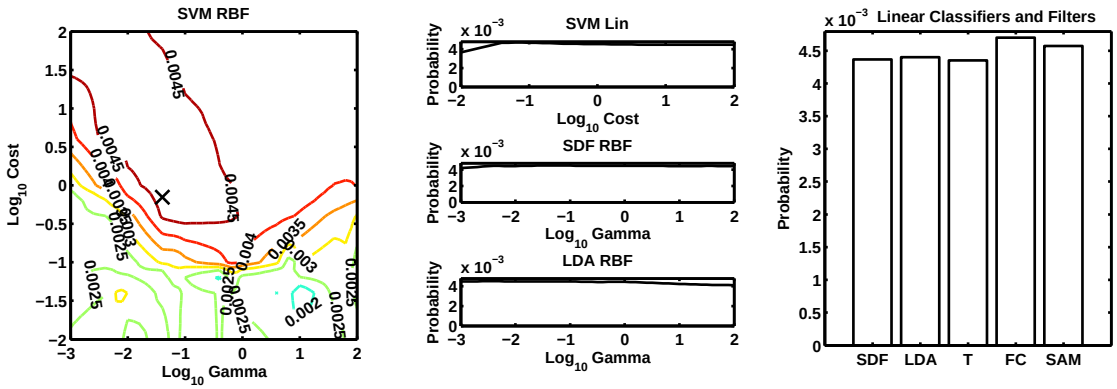
probabilities initialized using the randomly selected knowledge does not have distinct regions of biological relevance. In other words, there is not a set of feature ranking metrics that appear to stand out as the most biologically relevant given the initial knowledge. This implies that, as expected, none of the ranking metrics can correctly identify the initial biomarkers. As we detect and validate new biomarkers, the algorithm probabilities begin to form distinct regions that correspond to those in **Figure 27**.

Figure 29 is an example of the evolution of ranking metric probabilities when we introduce knowledge and update probabilities using the maximum *a posteriori* approach. The probabilities displayed in the first (**Figure 29(a)**), fifth (**Figure 29(b)**), and tenth (**Figure 29(c)**) iterations are posterior probabilities. The regions that emerge as the most biologically relevant are similar to those of the maximum likelihood method (**Figure 27**). However, we also see that regions of the linear SVM parameter space as well as the fold change and SAM methods also emerge as more biologically relevant compared to other feature ranking metrics. Despite incorrect initial knowledge, the maximum *a posteriori* approach to ranking metric selection still efficiently identifies biologically relevant regions of the parameter space within a few iterations (**Figure 30**).

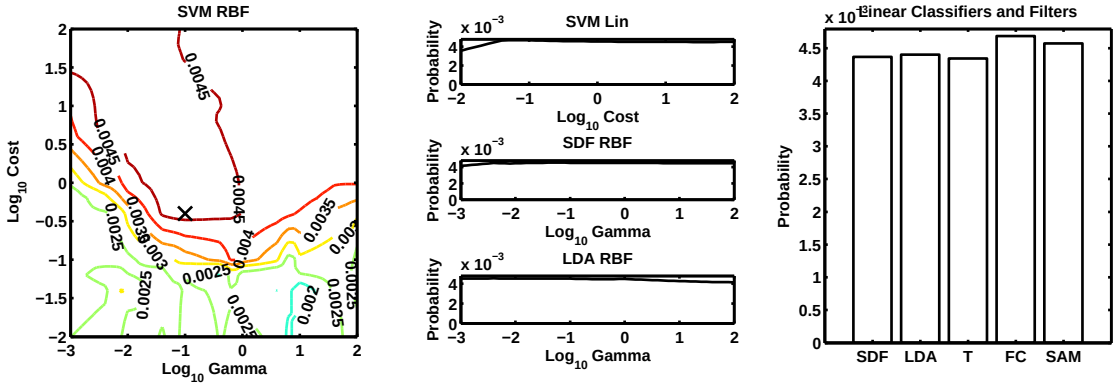
Figure 31 and Figure 32 are further illustrations of the advantage of using the Bayesian algorithm selection method in situations where initial knowledge may not be reliable. Although the maximum likelihood algorithm identifies a few regions of higher biological relevance, these regions are not distinctly different from other regions. In contrast, the Bayesian selection method not only identifies these regions, but the probabilities of these regions are distinctly different from other, non-biologically relevant regions.



(a) Iteration 1, Maximum Likelihood, Initial Knowledge

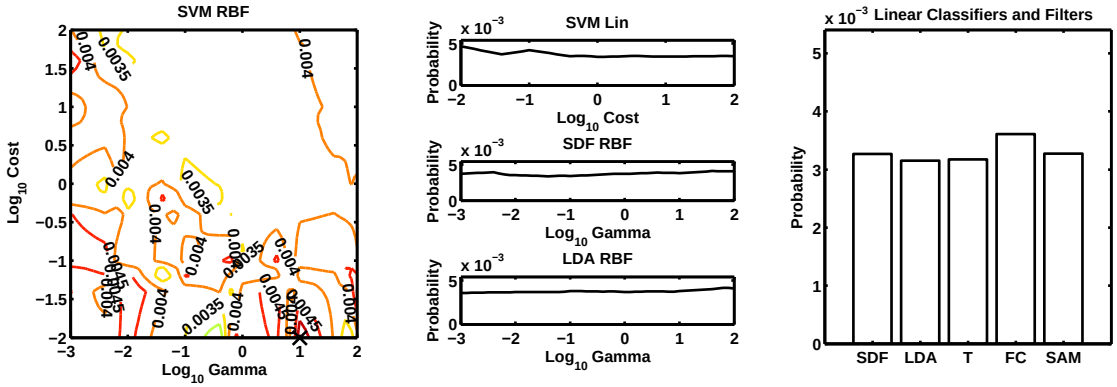


(b) Iteration 5, Maximum Likelihood, Initial Knowledge +5 Validated Biomarkers

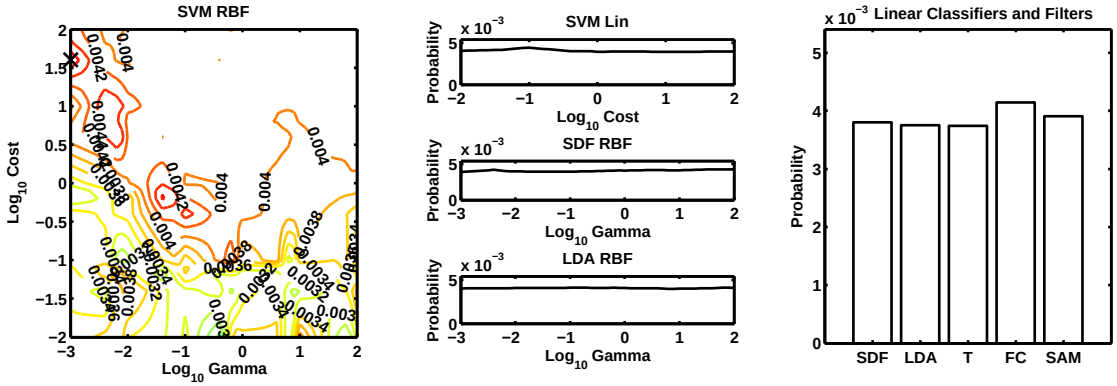


(c) Iteration 10, Maximum Likelihood, Initial Knowledge +10 Validated Biomarkers

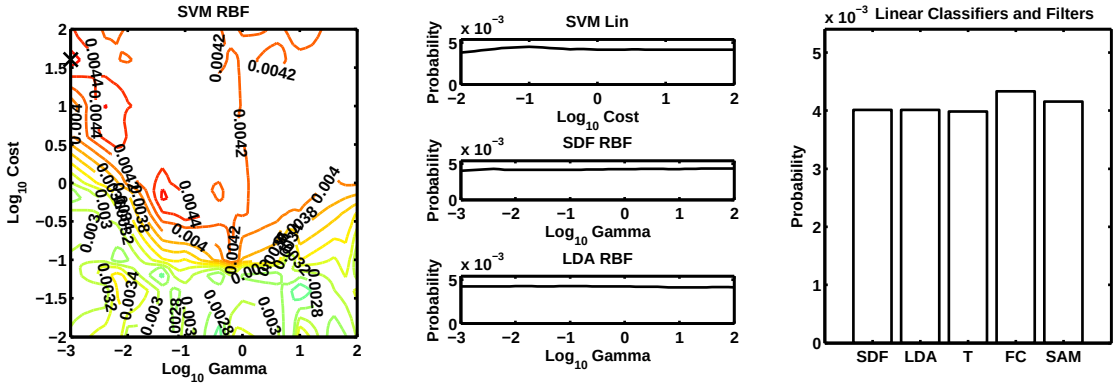
Figure 27: Evolution of ranking metric probabilities using the maximum likelihood algorithm. There is very little difference between the iteration 1 (**27(a)**), iteration 5 (**27(b)**), and iteration 10 (**27(c)**) iteration as new knowledge is added. However, the "optimal", or algorithm with maximum likelihood changes slightly (black X) within the space of SVM radial basis wrapper-based feature selection methods (left panel). Probabilities of common filter-based feature selection methods (right panel: T-test, fold change, and SAM) are not significantly different from those of wrapper based methods.



(a) Iteration 1, Maximum Likelihood, Random Initial Knowledge

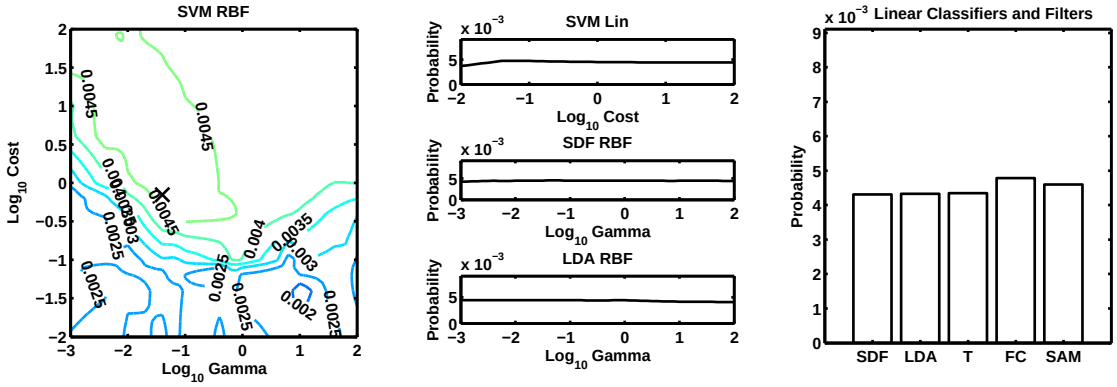


(b) Iteration 5, Maximum Likelihood, Random Initial Knowledge +5 Validated Biomarkers

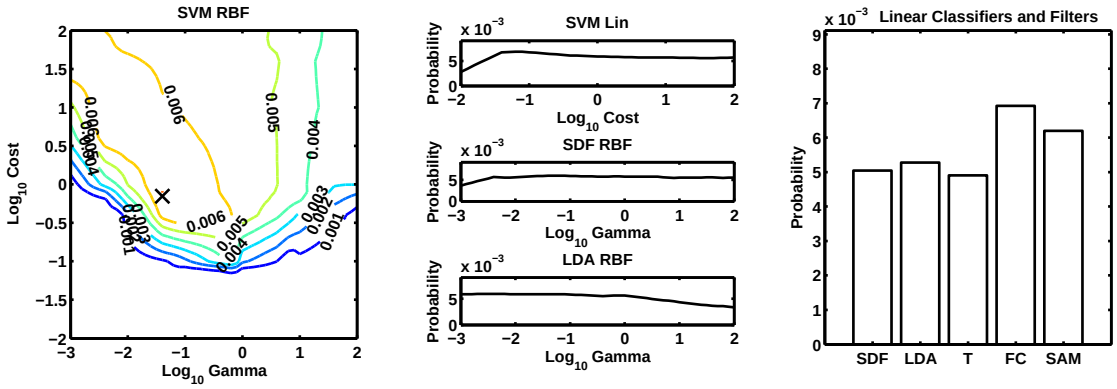


(c) Iteration 10, Maximum Likelihood, Random Initial Knowledge +10 Validated Biomarkers

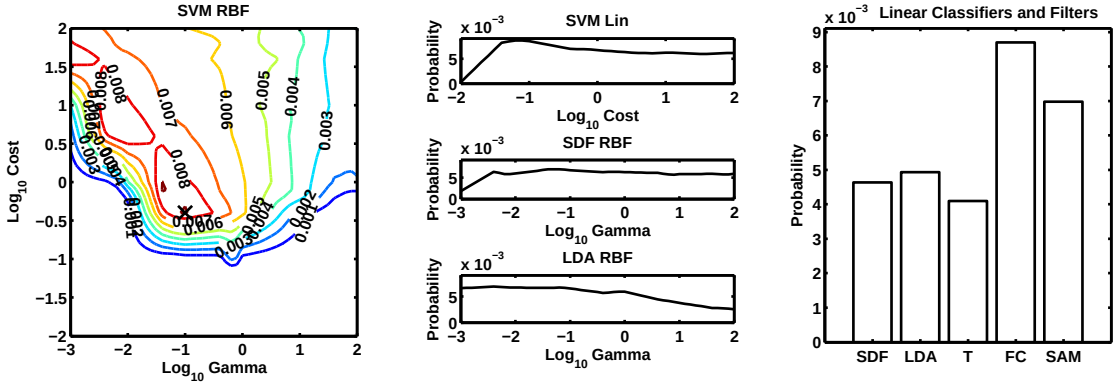
Figure 28: Evolution of ranking metric probabilities using the maximum likelihood algorithm with incorrect initial knowledge. Beginning with randomly selected initial knowledge, there is very little distinction between the different ranking metrics (**28(a)**). As knowledge accumulates, resulting in a mixture of incorrect and correct knowledge, some of the ranking metrics emerge and distinguish themselves from other metrics in terms of biological relevance after 10 iterations (**28(c)**). Probabilities of common filter-based feature selection methods (right panel: T-test, fold change, and SAM) are not significantly different from wrapper based methods.



(a) Iteration 1, Maximum *A Posteriori*, Initial Knowledge

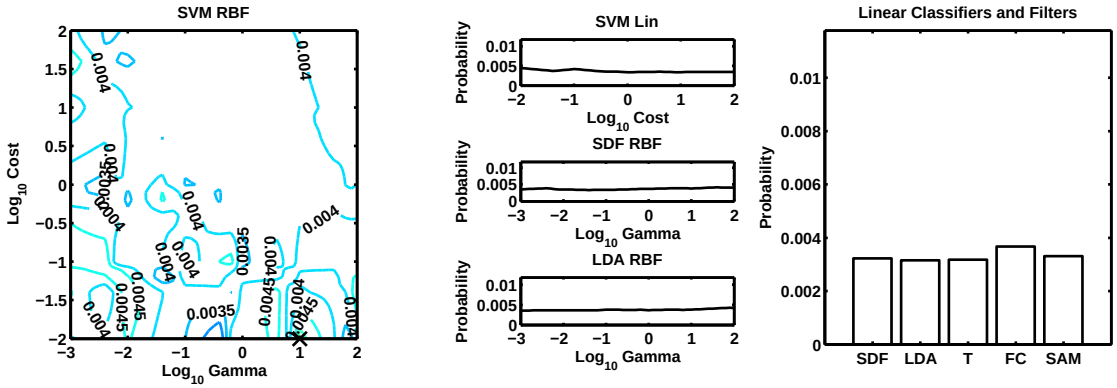


(b) Iteration 5, Maximum *A Posteriori*, Initial Knowledge +5 Validated Biomarkers

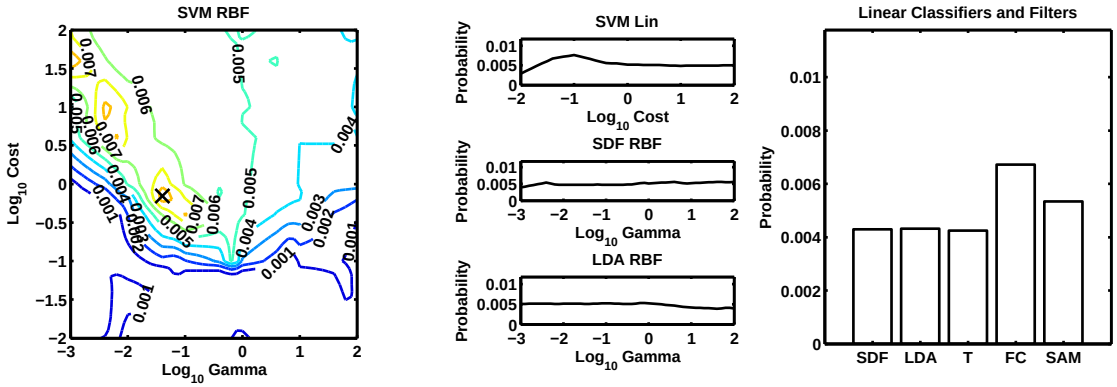


(c) Iteration 10, Maximum *A Posteriori*, Initial Knowledge +10 Validated Biomarkers

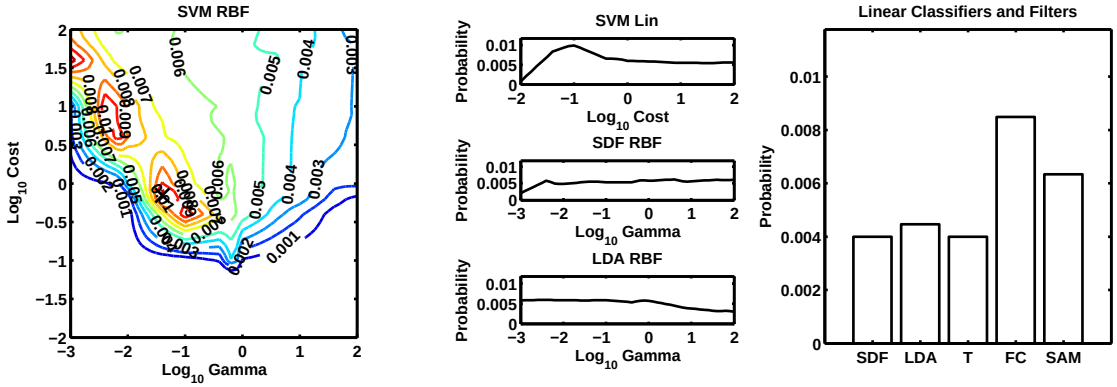
Figure 29: Evolution of the posterior probability of ranking metrics as knowledge is introduced. The space of ranking metrics considered here include the radial basis SVM (left panel), linear SVM (middle panel, top row), radial basis SDF (middle panel, middle row), radial basis linear discriminant (LDA, middle panel, bottom row), and linear SDF, linear LDA, and common filtering methods. Beginning with initial knowledge (**29(a)**) to 10 iterations (**29(c)**), the maximum *a posteriori* method clearly identifies ranking metrics that are biologically relevant.



(a) Iteration 1, Maximum *A Posteriori*, Random Initial Knowledge

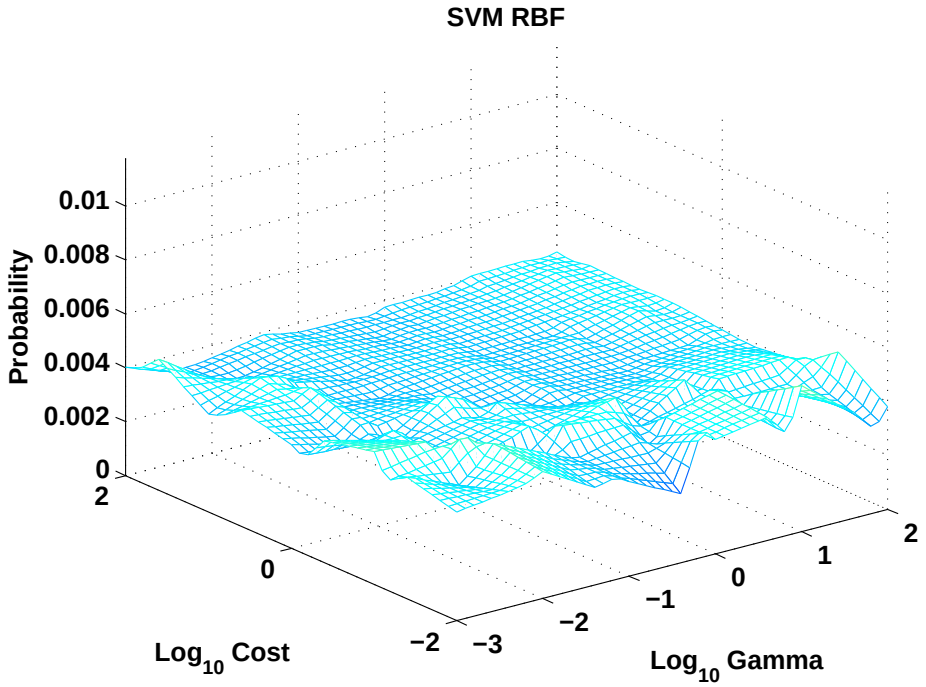


(b) Iteration 5, Maximum *A Posteriori*, Random Initial Knowledge +5 Validated Biomarkers



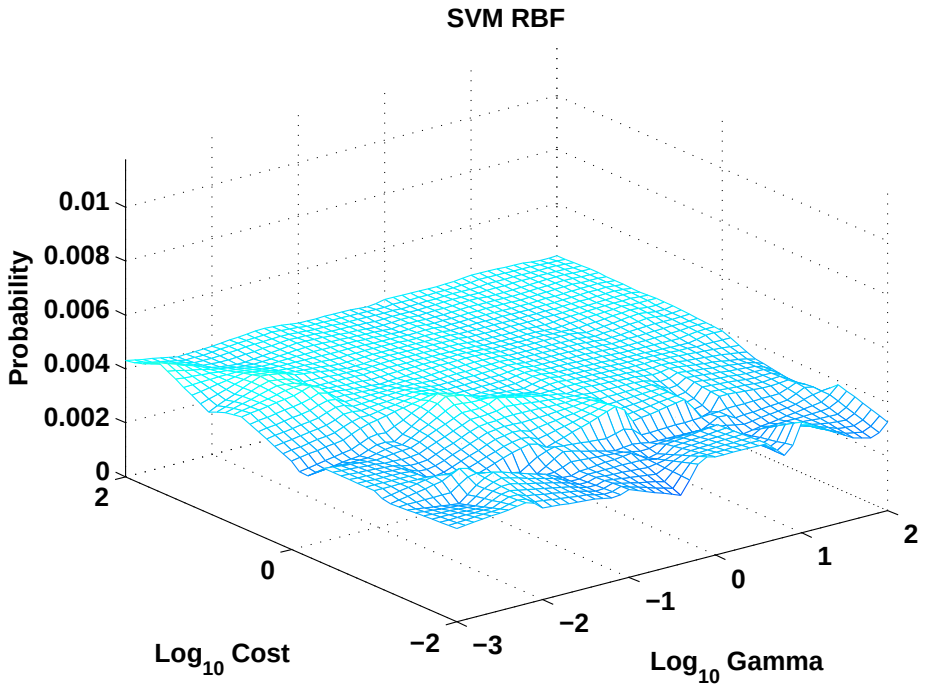
(c) Iteration 10, Maximum *A Posteriori*, Random Initial Knowledge +10 Validated Biomarkers

Figure 30: Evolution of the posterior probability of ranking metrics as knowledge is introduced and beginning with randomly selected initial knowledge. Despite the incorrect initial knowledge, or prior, the maximum *a posteriori* method quickly identifies regions of biological relevance after 10 iterations (**30(c)**). The space of ranking metrics considered here include the radial basis SVM (left panel), linear SVM (middle panel, top row), radial basis SDF (middle panel, middle row), radial basis linear discriminant (LDA, middle panel, bottom row), and linear SDF, linear LDA, and common filtering methods.

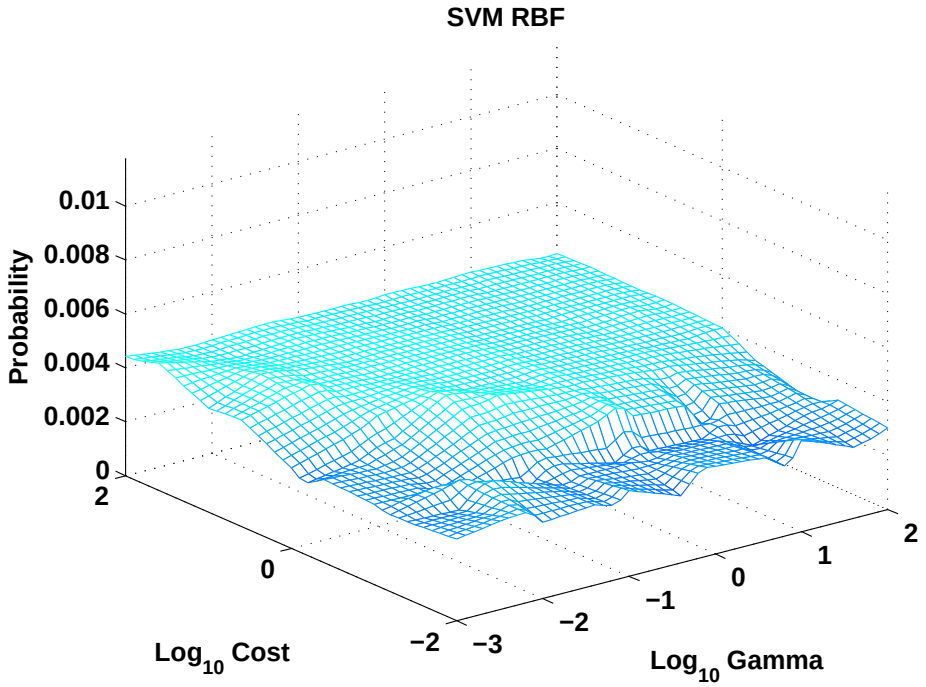


(a) SVM ranking metric probability surface after 1 iteration of the maximum likelihood method using random initial knowledge.

Figure 31: Evolution of the probability surface of the radial basis SVM ranking metrics using the maximum likelihood method and randomly selected initial knowledge. Some regions appear as more biologically relevant but are not significantly more so than the other regions. There is very little difference between probability surfaces after iteration 1 (**31(a)**, iteration 5 (**31(b)**, next page), and iteration 10 (**31(c)**, next page).

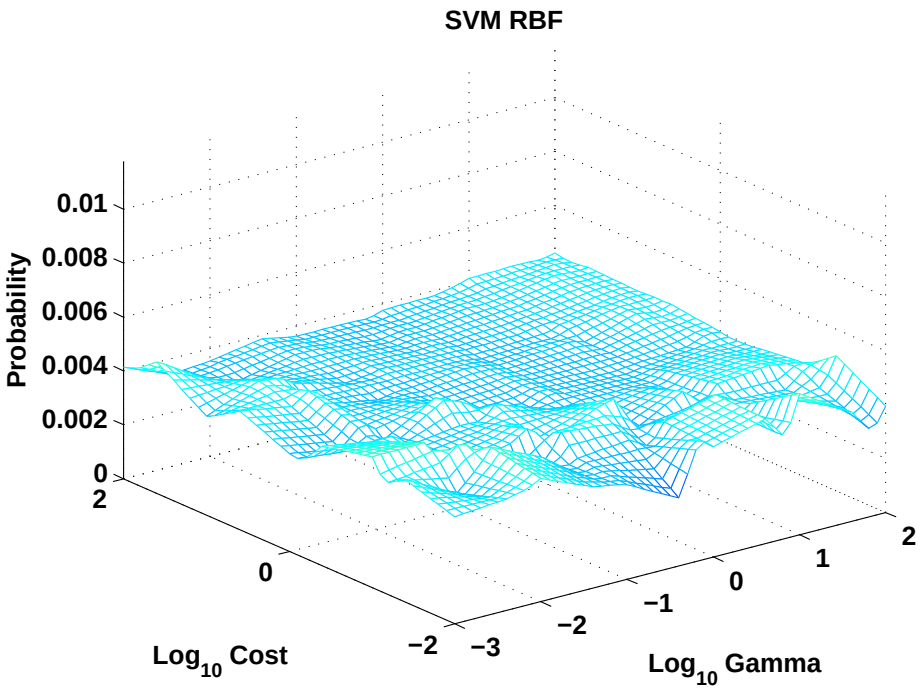


(b) SVM ranking metric probability surface after 5 iterations of the maximum likelihood method using random initial knowledge.



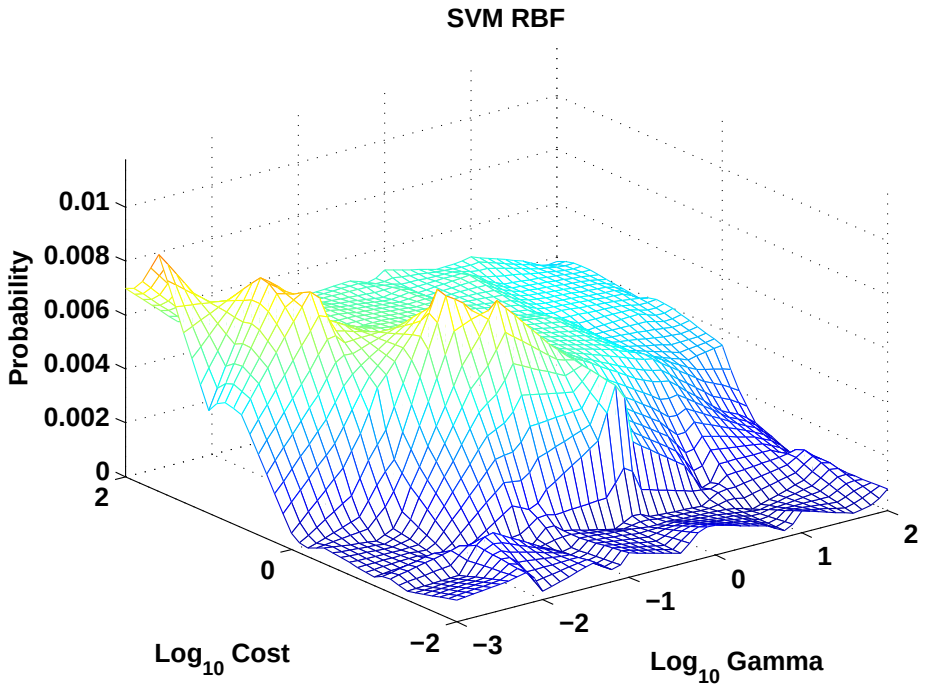
(c) SVM ranking metric probability surface after 10 iterations of the maximum likelihood method using random initial knowledge.

Figure 31 parts (b) and (c). Figure part (a) and full caption on the previous page.

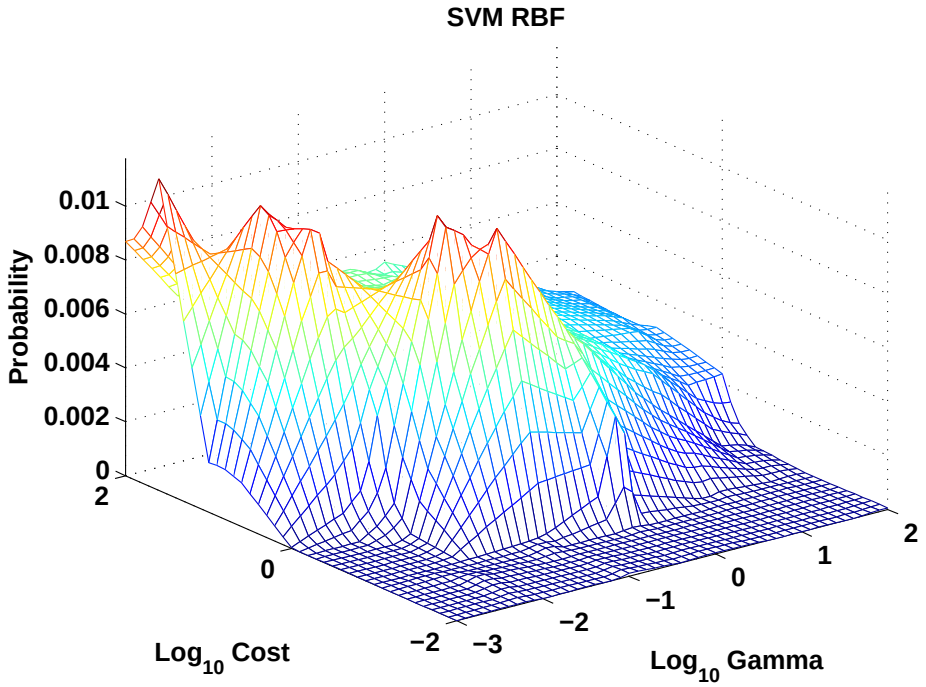


(a) SVM ranking metric probability surface after 1 iteration of the maximum *a posteriori* method using random initial knowledge.

Figure 32: Evolution of the radial basis SVM probability surface using the maximum *a posteriori* algorithm and randomly selected initial knowledge. Within 10 iterations (iteration 1 (**32(a)**), iteration 5 (**32(b)**, next page), and iteration 10 (**32(c)**, next page)), several regions appear to be distinctly more biologically relevant than other regions.



(b) SVM ranking metric probability surface after 5 iterations of the maximum *a posteriori* method using random initial knowledge.



(c) SVM ranking metric probability surface after 10 iterations of the maximum *a posteriori* method using random initial knowledge.

Figure 32 parts (b) and (c). Figure part (a) and full caption on the previous page.

4.3.4 Identifying and Validating Novel Biomarkers

Using all knowledge from literature and the first round of qRT-PCR for the CC and ONC/CHR comparison (endpoint A1 and A2), we optimize the ranking metric and select the top genes that have not been previously validated and that have estimated classification errors of less than 5% (**Table 13**). We can link a few of these genes directly to previous literature pertaining to renal cancer. For example, CXCR4 has been linked to kidney cancer. Using qRT-PCR, Schrader *et al.* shows that this gene is over-expressed in kidney cancer tissue compared to normal kidney tissue [123]. IGFBP3 and KLF10 have also been linked to renal cell carcinoma [117, 65]. Validation of these genes using qRT-PCR may yield additional knowledge to iteratively refine the biomarker selection process. However, since we want to primarily focus on the methodology here, we reserve the actual validation of these results for a future study.

Table 13: Proposed list of renal cancer genes for further qRT-PCR validation.

Gene Symbol	Estimated Error
ACLY	0
CXCR4	0.013907
C4A,C4B	0.0187
FLNA	0.019903
PMP22	0.023798
PFKFB3	0.026506
KLF10	0.027801
PRG1	0.03003
LGALS1	0.030617
PCCB	0.03274
TMSB10	0.034201
HCLS1	0.034415
ACTA2	0.039398
IGFBP3	0.040989
NFKBIA	0.042332
CD44	0.049095
IER3	0.049571

4.4 Conclusion

We have shown that biomarker identification by feature ranking benefits from knowledge integration at key points. Using this knowledge–whether from clinical observations, laboratory experiments, or existing literature–we can intelligently choose an optimal ranking metric for a specific gene expression dataset. The use of an optimal metric for ranking and identifying novel biomarkers reduces the number of false discoveries, increases the number of true discoveries, reduces the required time for validation, and increases the overall efficiency of the process. There are two methods for integrating knowledge: maximum likelihood and maximum *a posteriori* estimation. Both methods result in a similar increase in biomarker detection efficiency. However, the Bayesian inference method tends to be more robust to noise due to less reliable knowledge.

The results of our simulations indicate that knowledge integration improves biomarker selection for clinical microarray data. Although this study assumes independent gene expression, the method is general and we can use it to rank combinatorial gene expression data as well. Furthermore, we test this method using only a limited set of wrapper-based feature ranking metrics and common filter methods such as the t-test, fold change, and SAM. However, it is easily expandable to encompass a larger variety of metrics. We hope that the proposed method will impact biomarker identification practices and improve the effectiveness of resulting clinical applications.

CHAPTER V

IMPROVING CLINICAL PREDICTION USING BIOLOGICAL KNOWLEDGE

5.1 Introduction

We have seen that knowledge integration improves the biological relevance of feature selection [106]. However, we also want to know if biological knowledge can improve the accuracy of clinical prediction. Biomarker identification improves our understanding of the underlying biological mechanisms of disease. But it is unclear whether the biological relevance of a particular biomarker implies that it may also be used as an accurate predictor. In this chapter, we perform a systematic study of clinical predictors to determine whether the integration of biological knowledge into the feature selection process improves prediction performance.

The steps involved in building clinical predictors include (1) feature selection, (2) prediction performance estimation, and (3) prediction performance evaluation on external data. Step two is not necessary for building the final model. However, this step is essential in order to assess the performance of a predictor on future samples. The process of building a predictive model may be carried out from a purely data-driven perspective, in which the biological relevance of the model–e.g., the features selected for the model–is not verified. Indeed, such a perspective may result in a wide variety of models that are not biologically related but perform equally well (or equally poorly). We expect the variance in model performance to be greatly affected by the feature selection method. In this study, we examine several feature selection methods and compare some common filter-based methods to a method that is knowledge-driven rather than purely data-driven. We hope to determine whether

integrating biological knowledge in the feature selection step results in more consistent and better performing predictive models.

Clinical predictors must be subject to heavy scrutiny before final application in clinical practice. However, we can only determine the impact of a predictor after acquiring an adequate number of patient samples and comparing prediction results to final clinical outcomes. The ultimate recommendation of a classification model is based on an estimate of future prediction performance. Unfortunately, this estimate depends on a subset of the total patient population, which may lead to a biased estimate. Biased estimates are generally optimistic due to data over-fitting–i.e., the prediction model becomes specialized to the patient sub-population and cannot generalize to the whole population. Thus, recommended models that are biased will lead to poor prediction on future samples. Proper cross validation procedures on the patient sub-population, or training data, will generally predict external validation performance [132].

5.2 Methods

5.2.1 Microarray Data

We build clinical predictive models from three different datasets using several feature selection methods and classifiers (**Table 14**). The datasets include renal cancer, prostate cancer, and breast cancer, each from public sources. We use two independent datasets for each cancer in order to compute an unbiased estimate of predictive model performance. The endpoint of a dataset refers to the specific medical condition we wish to predict. For example, the clinical goal of endpoint A of the renal cancer data is to develop a predictor that can classify patient samples into either the clear cell carcinoma subtype or a combination of the oncocytoma and chromophobe subtypes. Despite the very small sample size of one of the renal cancer datasets, we include this dataset to assess the effect of small sample size on prediction performance. The

Table 14: Clinical microarray datasets for knowledge-guided biomarker identification. For each endpoint, we use two microarray datasets to examine the effect of knowledge-guided feature selection. Each dataset varies in terms of clinical endpoint as well as sample size. Knowledge genes used to assess biomarker detection efficiency are identified from literature as well as from qRT-PCR experiments.

Dataset Code	Endpoint Code	# Knowledge Genes/Probesets	Endpoint Description	Dataset 1		Dataset 2	
				# P	# N	# P	# N
Renal	A	21	Clear Cell vs Onco./Chromo.	13	7	32	18
	B	12	Clear Cell vs Papillary	13	5	32	11
Prostate	C	6	Tumor vs Norm. Adj. Tissue	52	50	61	63
Breast	D	8	pCR vs RD	21	60	12	37

expected result is that the cross validation step will overestimate model performance. We are also interested in the effect of knowledge integration on prediction performance in small sample situations. The goal of endpoint B is to classify patients into the clear cell and papillary subtypes [124, 67]. Endpoint C examines prostate cancer with the goal of identifying molecular markers that can distinguish between prostate tumor and normal adjacent tissue from the same patient [133, 19]. The prostate cancer datasets have relatively large sample sizes compared to the renal cancer data. Endpoint D examines breast cancer treatment outcomes. In the original study from which this data was derived, the authors used two datasets to assess prediction accuracy of several classifiers [53].

5.2.2 Estimating Predictive Performance Using Cross Validation

We build clinical predictors and assess their predictive performance using a full cross validation procedure that is designed to mimic clinical scenarios in which the validation data is unknown or does not even exist during the training process. In clinical practice, we hope to have a validated predictive model before receiving any new

samples for classification. Consequently, the model building and selection procedure should be completely separate from the validation procedure. In order to estimate the performance of a potential predictor, we need to split the samples into training and testing groups. All steps involved in determining the model–e.g. feature and parameter selection–should be applied to the training data. Once the model parameters have been established, the model performance may be evaluated on the testing data. The selection of samples for the training and testing data may also introduce a bias. Therefore we recommended to create multiple partitions by using either cross validation or bootstrap. We use 10 iterations of stratified 5-fold cross validation. Within each iteration, we compute and average five validation scores. We then compute the average and standard deviation of the resulting 10 validation scores and use these numbers as an estimate of model performance. **Figure 33** is an example of this process using 3-fold cross validation.

At this point, we have a performance estimate for each model via cross validation. We then perform feature selection on the entire dataset and build a single classifier using these features. This classifier is tested on an external dataset whose samples were never used during the cross validation process. Assuming that this external dataset is similar to the original dataset, the estimated predictive performance computed using cross validation should be similar to the performance calculated from external validation. The performance of the selected model should be high, but the ultimate criterion of performance, of course, is that external validation is also accurate.

In terms of the microarray data, we perform full cross validation using datasets A1, B1, C1, and D1 in order to estimate predictive performance. Using the same modeling parameters as the cross validation step, we then apply these models to train with the full A1, B1, C1, and D1 datasets and predict samples from the A2, B2, C2, and D2 datasets. As such, each combination of modeling factors includes an internal cross validation (CV) estimate of performance as well as an external "blind"

Table 15: Parameters for wrapper-based feature selection methods.

Classifier	Kernel	Parameters
SVM	Linear	C:0.01,0.04,0.07,0.1,0.4,0.7,1,4,7,10,40,70,100
	Gaussian	C:0.01,0.04,0.07,0.1,0.4,0.7,1,4,7,10,40,70,100 γ:0.001,0.004,0.007,0.01,0.04,0.07,0.1,0.4,0.7, 1,4,7,10,40,70,100
SDF	Linear	N/A
	Gaussian	γ:0.001,0.004,0.007,0.01,0.04,0.07,0.1,0.4,0.7, 1,4,7,10,40,70,100
LDA	Linear	N/A
	Gaussian	γ:0.001,0.004,0.007,0.01,0.04,0.07,0.1,0.4,0.7, 1,4,7,10,40,70,100

validation (EV) score. We expect that the CV estimate and the EV score should be comparable if the corresponding model does not over-fit to the training data. In order to test the robustness of each model to changes in sample population, we swap the training and testing data and repeat the cross validation and validation procedures. For example, we perform full cross validation on the A2, B2, C2, and D2 datasets, then validate these models using the A1, B1, C1, and D1 datasets.

5.2.3 Feature Selection and Biological Relevance

We test several feature selection methods, including the commonly used t-test, fold change, and significance analysis of microarrays (SAM) [141]. Additionally, we also test several wrapper based methods using the support vector machine (SVM), signed distance function (SDF), and linear discriminant classifiers (LDA) [28, 6]. For each classifier, we use the linear and radial basis kernel functions varied over several parameters (**Table 15**). We rank features by estimating their classification error using 100 iterations 0.632+ bootstrap [10, 37]. In total, we examine 258 different feature selection methods and vary feature sizes from 5 to 30 in steps of 5 (i.e., 5,10,15,20,25,30).

For each of the four datasets, we identify several previously validated biomarkers for use as reference knowledge in order to assess the biological relevance of each

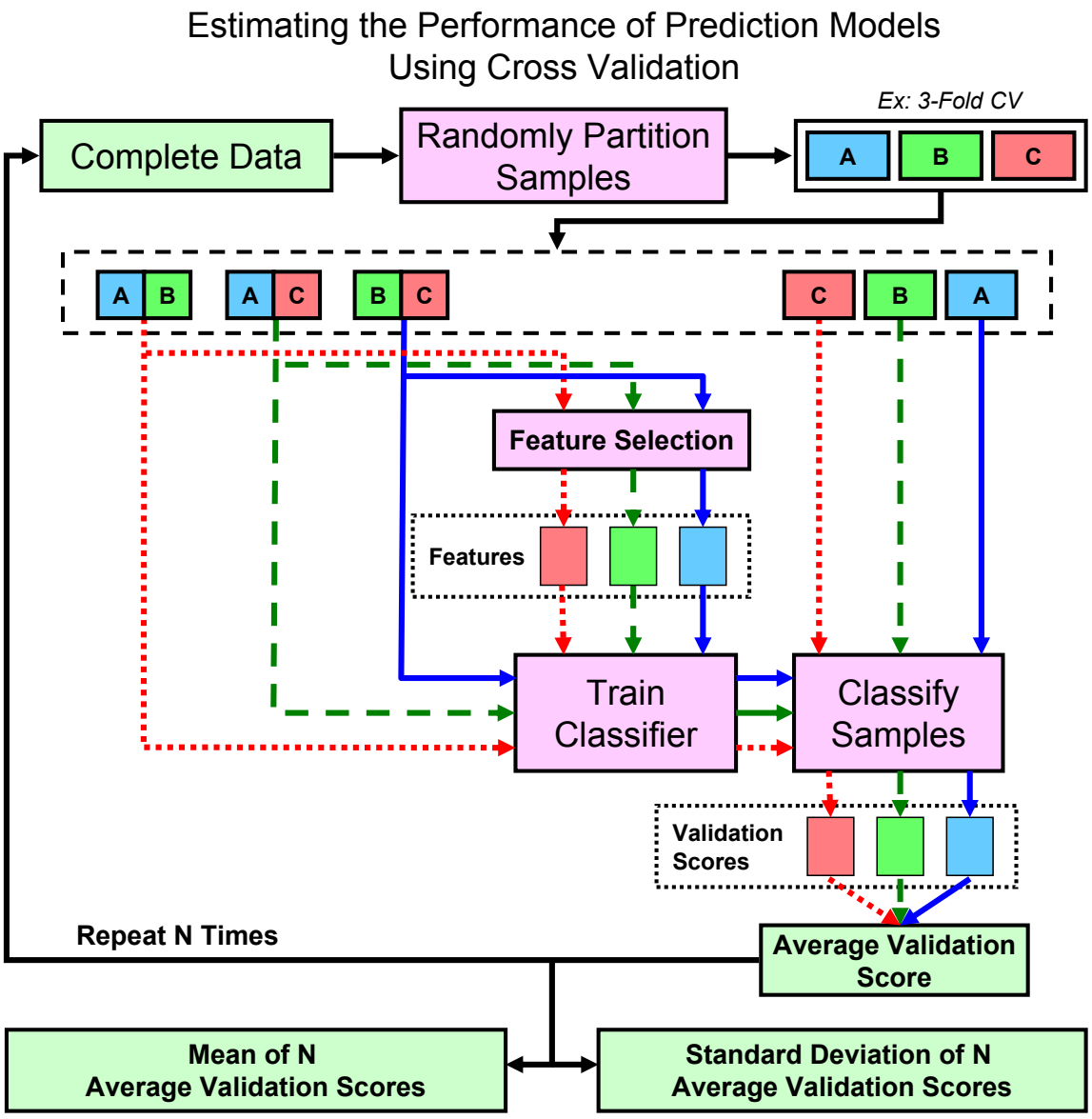


Figure 33: Assessing clinical predictor performance using full cross validation. In order to compute an unbiased estimate of classifier performance on future data, we need to perform a full cross validation in which all steps involved in the model building process–feature selection and classifier parameter tuning–must be performed within the cross validation.

feature selection method. A feature selection method, θ, assigns to each gene, i, a score based on its differential expression using a function $h_\theta(\vec{d_i})$, where $\vec{d_i}$ represents expression values of the gene across all samples in the dataset and $i = 1 \ldots m$. We assume that lower ranking scores indicate higher differential expression and that all scores are constrained to be within the interval [0,1]. We define $G_k = \{g_1, g_2, \ldots, g_3\}$ as the set of k relevant biomarkers such that elements of the set $\{h_\theta(\vec{d_i}) : i \in G_k\}$ are generally smaller than those of $\{h_\theta(\vec{d_j}) : j \in G_k\}$. Then we can define the following function as the biological relevance of a feature ranking method, θ:

$$\phi(G_k, \theta) = \frac{1}{k(m-k)} \sum_{i \in G_k} \sum_{j \notin G_k} I(h_\theta(\vec{d_i}) < h_\theta(\vec{d_j})) \tag{20}$$

where $I(x)$ is the indicator function that evaluates to one when x is true and zero when x is false. **Equation 20** is equivalent to the area under an ROC curve [93]. This notation is similar to that used in a previous study that examined the biological relevance of feature ranking [106]. We identify reference knowledge for each endpoint using both literature and qRT-PCR validation (**Table 16**, **Table 17**, **Table 18**, and **Table 19**).

5.2.4 Classifiers

We use three classifiers for both the wrapper-based feature selection methods and the final predictive classifiers–support vector machines (SVM) [28], signed distance function (SDF) [6], and linear discriminant analysis (LDA). Each classifier may be applied in either linear or kernelized form. We use the Gaussian kernel and adjust the variance parameter to achieve different levels of non-linear classification (**Table 20**). In total, we use 30 different classifiers in varying degrees of complexity.

5.2.5 Classification Performance Metrics

There are several method for measuring the performance of a classifier. Each metric examines a different aspect of a classifier. We use three metrics: accuracy, area

Table 16: Genes validated as differentially expressed between CC and ONC/CHR renal tumor subtypes from various knowledge sources.

Gene Symbol	Knowledge Source	Validation Method
CA9	Chen, Clin Cancer Res, 2005	qRT-PCR
CLCNKB	Chen, Clin Cancer Res, 2005	qRT-PCR
DEFB1	Schuetz, J Mol Diagn, 2005	qRT-PCR, IHC
LRP2	Schuetz, J Mol Diagn, 2005	qRT-PCR, IHC
PVALB	Chen, Clin Cancer Res, 2005	qRT-PCR
STC1	Phan, Pac Symp Biocomput, 2009	qRT-PCR
SLC25A4	Phan, Pac Symp Biocomput, 2009	qRT-PCR
CFTR	Phan, Pac Symp Biocomput, 2009	qRT-PCR
PDHA1	Phan, Pac Symp Biocomput, 2009	qRT-PCR
PFKM	Phan, Pac Symp Biocomput, 2009	qRT-PCR
NNMT	Phan, Pac Symp Biocomput, 2009	qRT-PCR
CP	Phan, Pac Symp Biocomput, 2009	qRT-PCR
CFB	Phan, Pac Symp Biocomput, 2009	qRT-PCR
COX5A	Phan, Pac Symp Biocomput, 2009	qRT-PCR
BAG1	Phan, Pac Symp Biocomput, 2009	qRT-PCR
LY6E	Phan, Pac Symp Biocomput, 2009	qRT-PCR
CD99	Phan, Pac Symp Biocomput, 2009	qRT-PCR
AKAP12	Phan, Pac Symp Biocomput, 2009	qRT-PCR
ACAT1	Phan, Pac Symp Biocomput, 2009	qRT-PCR
SPTBN2	Phan, Pac Symp Biocomput, 2009	qRT-PCR
GOT1	Phan, Pac Symp Biocomput, 2009	qRT-PCR

Table 17: Genes validated as differentially expressed between CC and PAP renal tumor subtypes from an in-house knowledge source.

Gene Symbol	Knowledge Source	Validation Method
STC1	In-House	qRT-PCR
NDUFA4L2	In-House	qRT-PCR
CA9	In-House	qRT-PCR
CP	In-House	qRT-PCR
ELF3	In-House	qRT-PCR
BST2	In-House	qRT-PCR
B3GNT4	In-House	qRT-PCR
GRB7	In-House	qRT-PCR
BAMBI	In-House	qRT-PCR
CCL20	In-House	qRT-PCR
CTSC	In-House	qRT-PCR
PECAM1	In-House	qRT-PCR

Table 18: Validated prostate cancer biomarkers identified from literature. These biomarkers are specific for distinguishing between prostate tumor and normal prostate tissue.

Gene Symbol	Source
SARDH	Sreekumar, Nature, 2009
AMACR	Luo, Cancer Research, 2002
HPN	Sardana, Clin Chem, 2008
MYC	Prowatke, British Journal of Cancer, 2007
FASN	Prowatke, British Journal of Cancer, 2007
FOLH1	Schlomm, Eur Urol, 2008

Table 19: Validated breast cancer biomarkers identified from literature. These biomarkers are specific for distinguishing between chemotherapy treatment outcome. Specifically, for pathologic complete response (pCR) to T/FAC treatment versus residual disease after a predefined period of time.

Gene Symbol	Source
MAPT	Rouzier, PNAS, 2005
KI67	Burcombe, Breast Cancer Res, 2006

Table 20: Classifier parameters for predictive model assessment.

Classifier	Kernel	Parameters
SVM	Linear	C:0.1,1,10,100
	Gaussian	C:0.1,1,10,100
		γ:0.0001,0.001,0.01,0.1
SDF	Linear	N/A
	Gaussian	γ:0.0001,0.001,0.01,0.1
LDA	Linear	N/A
	Gaussian	γ:0.0001,0.001,0.01,0.1

Table 21: Summary of modeling factors in systematic clinical prediction study.

Modeling Factor	Degrees of Freedom	Description
Clinical Endpoints	8 (including swap)	Renal (A1,A2,B1,B2) Prostate (C1,C2) Breast (D1,D2)
Feature Selection	258*	Filter: Fold Change, t-test, SAM Wrapper: SVM, SDF, LDA
Classifiers	30	SVM, SDF, LDA Linear and Gaussian Kernels
Feature Sizes	6	5,10,15,20,25,30

*We only considered 211 feature selection methods for endpoints D1 and D2

under the ROC curve (AUC), and Matthews Correlation Coefficient (MCC). Accuray and MCC are binary performance metrics, meaning that they do not consider the confidence, or probability of correct classification–samples are either correctly or incorrectly classified. AUC, on the other hand, takes into consideration, the degree of correct or incorrect classifications [8, 76]. Accuracy is highly sensitive to data prevalence. MCC attempts to correct the sensitivity of accuracy to data prevalence, but still behaves non-linearly for unbalanced data prevalence. AUC is invariant to data prevalence, but combines all classification thresholds. Ferri *et al.* conducted an extensive examination of several performance metrics [40]. For a more detailed description of accuracy, MCC, and AUC, refer to **Appendix D**.

5.2.6 Summary of Systematic Study

In total, we examine several modeling factors for four clinical endpoints, including swapped training and testing data (**Table 21**). The primary focus of the study is on the biological relevance of feature selection methods. However, we also vary the classifiers and feature sizes. We consider a total of 46,440 predictive models for each clinical endpoint (with the exception of the breast cancer endpoints, for which we only consider 37,980 models).

5.3 Results and Discussion

5.3.1 Data Batch Effect

Each of the four end points differs in terms of clinical focus and difficulty. These differences in data affect the performance of the resulting predictive models when tested on external validation data. However, there are other factors that can influence the reliability of model performance. These factors are primarily related to data quality and include sample size and batch effect. Both of these factors can bias the resulting predictive models, especially during the training phase, in which we estimate the performance of a predictor in order to select model parameters. Among the three datasets (four clinical endpoints), the renal cancer data from the study by Jones *et al.* has the largest batch effect between classes [67] (**Figure 34**). Furthermore, the batch effect also translates to a significant difference between the Jones data (endpoint A2 and B2) and the Schuetz renal cancer data (endpoint A1 and B1). We can see in **Figure 34** that there is a distinct separation of clusters between the CC_S (Schuetz Renal Cancer Clear Cell Samples) and CC_J (Jones Renal Cancer Clear Cell Samples) samples as well as between the CC_J and CHR_J (Jones Renal Cancer Chromophobe Samples) samples. We expect the batch effect within the Jones data to affect feature selection and, subsequently, predictive accuracy. For example, the batch effect will result in selection of a large number of differentially expressed genes purely due to differences between samples. Consequently, cross validation performance of models for the Jones data, endpoints A2 and B2, should be optimistic compared to the validation of these models on the Schuetz data, endpoints A1 and B1. The batch effect between endpoints A1 and B1 and endpoints A2 and B2 should also decrease validation performance.

We attempt to correct the batch effect between datasets using quantile normalization [7]. For example, prior to classifying samples in endpoint A2 using predictive

models created with samples from endpoint A1, we normalize all samples in endpoint A2 using samples from endpoint A1 as the reference distribution. **Figure 35**, **Figure 36**, **Figure 37**, and **Figure 38** illustrate this process. The red lines indicate the distribution of the normalized dataset. **Figure 35** and **Figure 36** also confirm the significant batch effect between the Jones data (end points A2 and B2) and the Schuetz data (end points A1 and B1). In addition to the differences between lab sources, the differences between these datasets may be attributed to the use of different probe-level summary normalization methods [62]. Because raw chip data are available for the Schuetz data, we compute probeset expression values using the MAS5 algorithm [58]. However, the raw chip data are not available for the Jones data. Instead we only have access to the probeset expression values computed using the dChip software package [78]. The batch effect within the prostate and breast cancer datasets are less prominent due to the availability of raw array data (**Figure 37** and **Figure 38**).

5.3.2 Estimating Performance of Predictive Models

Considering all models for each clinical endpoint, cross validation seems to predict external validation performance very poorly. This is especially true for the renal cancer data (endpoints A1 and B1) (**Figure 39**, green and red). Factors that contribute to poor concordance between cross validation and external validation are likely the small sample size, the variety of feature selection methods, and data batch effects. Cross validation is unreliable when the test sample for each fold consists very few samples. The set of 258 feature selection methods includes many wrapper based methods that may over-fit when estimating the predictive potential for individual biomarkers. The cross validation and external validation predictive performance of both the prostate cancer and breast cancer endpoints (C1 and D1) are more tightly distributed compared to the renal cancer endpoints (**Figure 39**, blue and magenta). This is likely

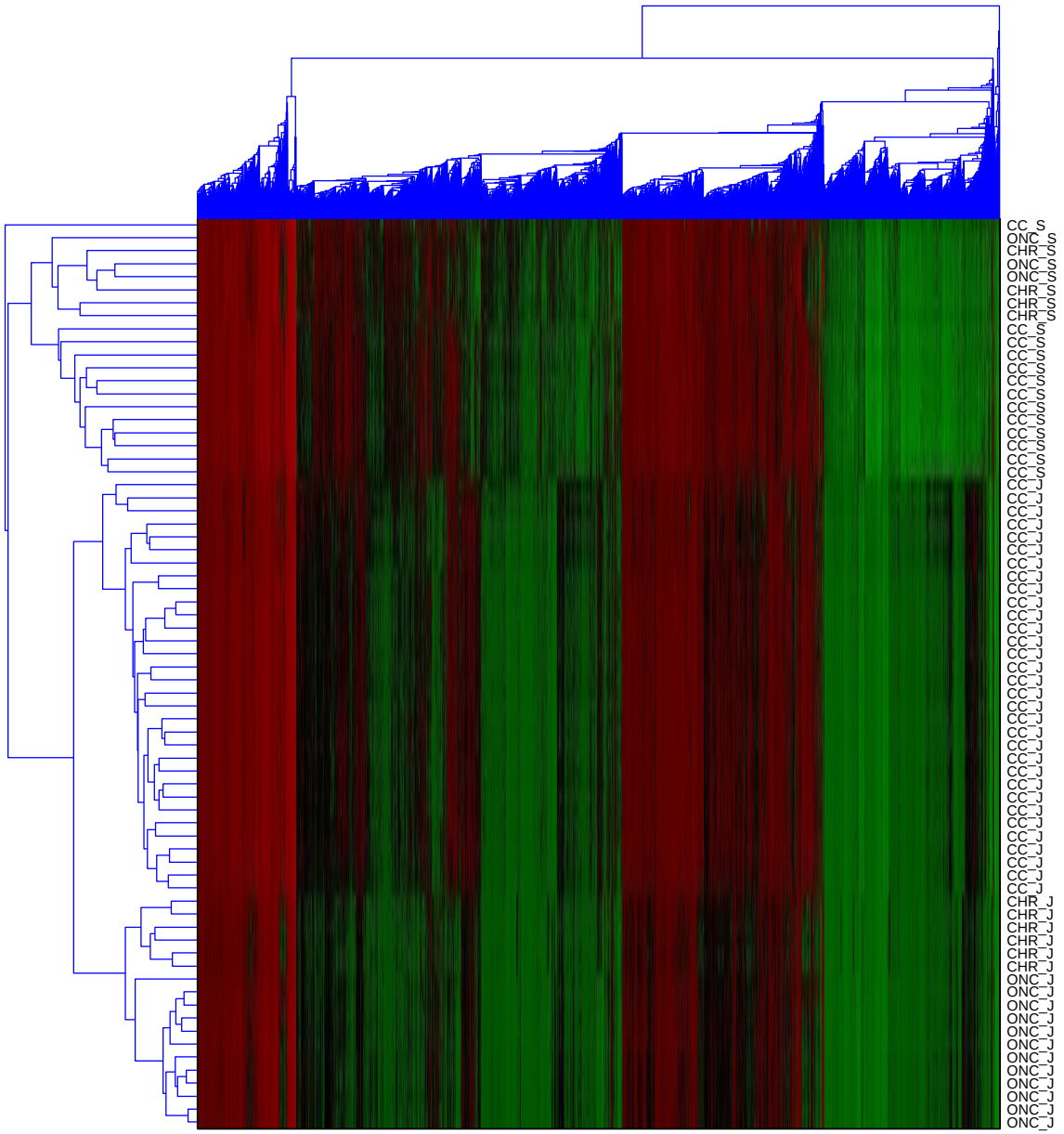


Figure 34: Batch effect in the renal cancer microarray data. Data endpoints A1 and A2 (as well as B1 and B2) are derived from two independent renal cancer studies. Samples for endpoint A1 were assayed using the HG-Focus Affymetrix chip while those for endpoints A2 were assayed using the HG-U133A chip. Using hierarchical clustering, there is a significant batch effect between the two datasets that should be addressed prior to building any predictive models. The CC_J, CHR_J, and ONC_J labels identify samples from the A2 endpoint while the CC_S, CHR_S, and ONC_S labels identify samples from the A1 endpoint. We also see a significant batch effect within the A1 endpoint between the CC_J samples and the group of CHR_J and ONC_J samples.

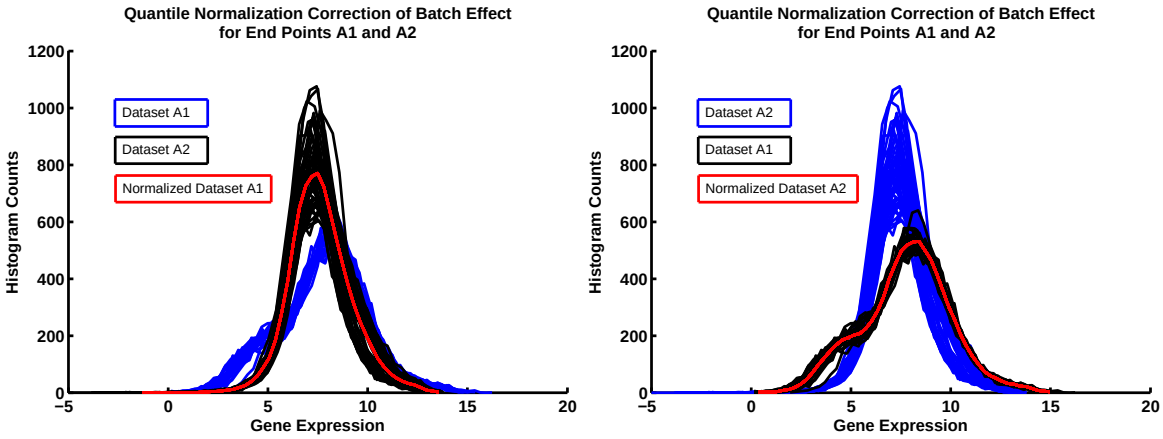


Figure 35: Batch effect between the A1 and A2 datasets, comparing renal cancer CC to ONC/CHR. Raw probe level data was not available for the A2 dataset. Therefore, we used gene expression values produced by different summary normalization software, resulting in a significant batch effect. Before classifying samples in the A1 dataset using predictive models created from the A2 dataset, we quantile normalize all samples in the A1 dataset using the A2 dataset as a reference distribution (left). Likewise, we normalize all samples in the A2 dataset using the A1 dataset as a reference distribution (right) before classifying samples in A2 using models created from A1.

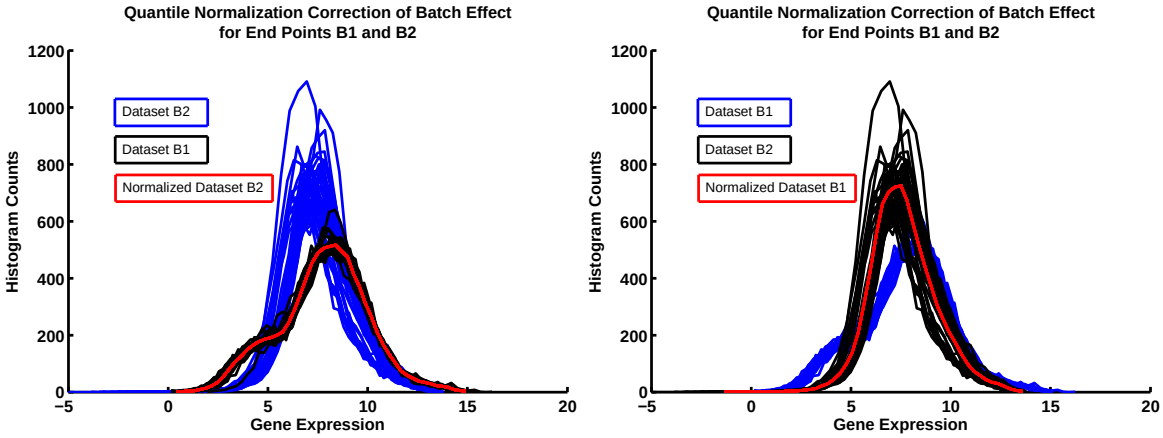


Figure 36: Batch effect between the B1 and B2 datasets, comparing renal cancer CC to PAP subtypes. Raw probe level data was not available for the B2 dataset. Therefore, we used gene expression values produced by different summary normalization software, resulting in a significant batch effect. Before classifying samples in the B1 dataset using predictive models created from the B2 dataset, we quantile normalize all samples in the B1 dataset using the B2 dataset as a reference distribution (left). Likewise, we normalize all samples in the B2 dataset using the B1 dataset as a reference distribution (right) before classifying samples in B2 using models created from B1.

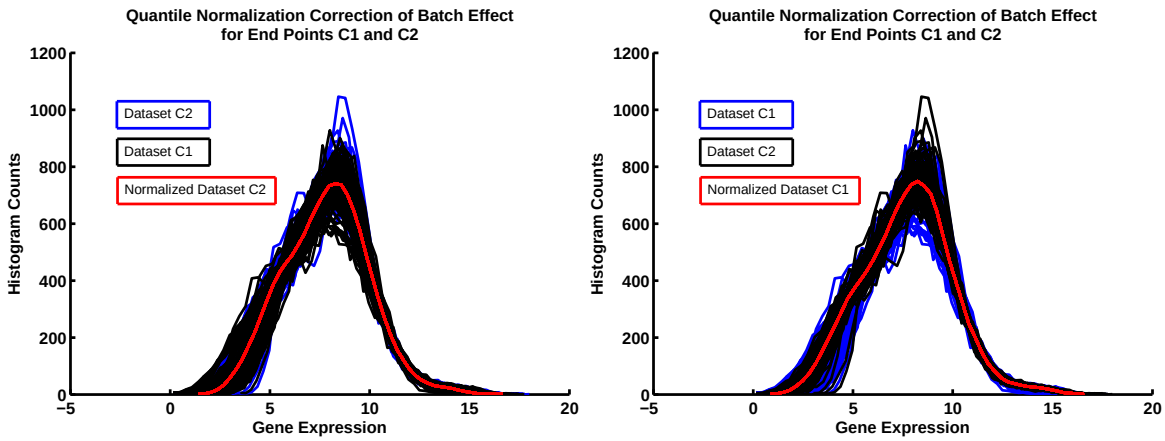


Figure 37: Batch effect between the C1 and C2 datasets, comparing prostate cancer tumor tissue to normal adjacent tissue. Compared to the renal cancer datasets, the batch effect in the prostate cancer datasets is minimal due to the availability of raw probe level information. All gene expression summarizations were computed using the same software. Before classifying samples in the C1 dataset using predictive models created from the C2 dataset, we quantile normalize all samples in the C1 dataset using the C2 dataset as a reference distribution (left). Likewise, we normalize all samples in the C2 dataset using the C1 dataset as a reference distribution (right) before classifying samples in C2 using models created from C1.

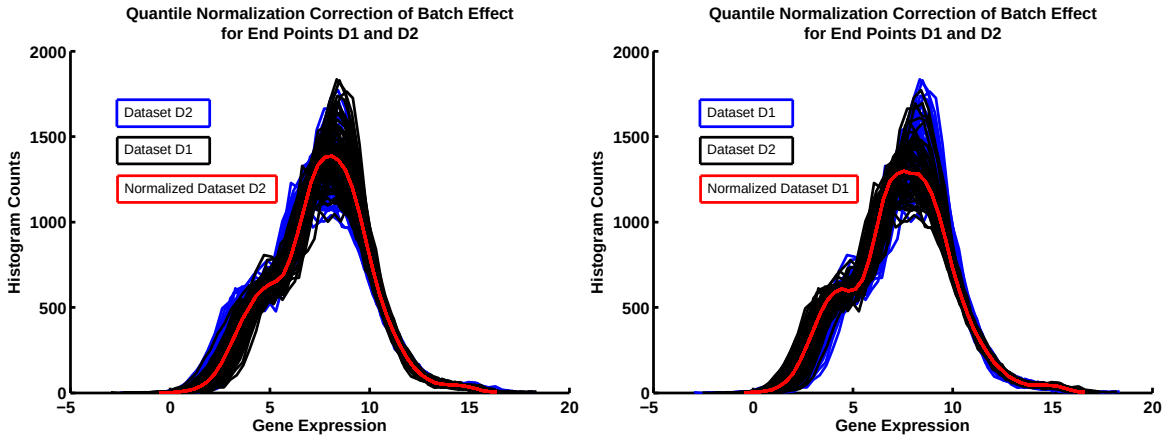


Figure 38: Batch effect between the D1 and D2 datasets, comparing breast cancer treatment outcomes. Compared to the renal cancer datasets, the batch effect in the breast cancer datasets is minimal due to the availability of raw probe level information. All gene expression summarizations were computed using the same software. Before classifying samples in the D1 dataset using predictive models created from the D2 dataset, we quantile normalize all samples in the D1 dataset using the D2 dataset as a reference distribution (left). Likewise, we normalize all samples in the D2 dataset using the D1 dataset as a reference distribution (right) before classifying samples in D2 using models created from D1.

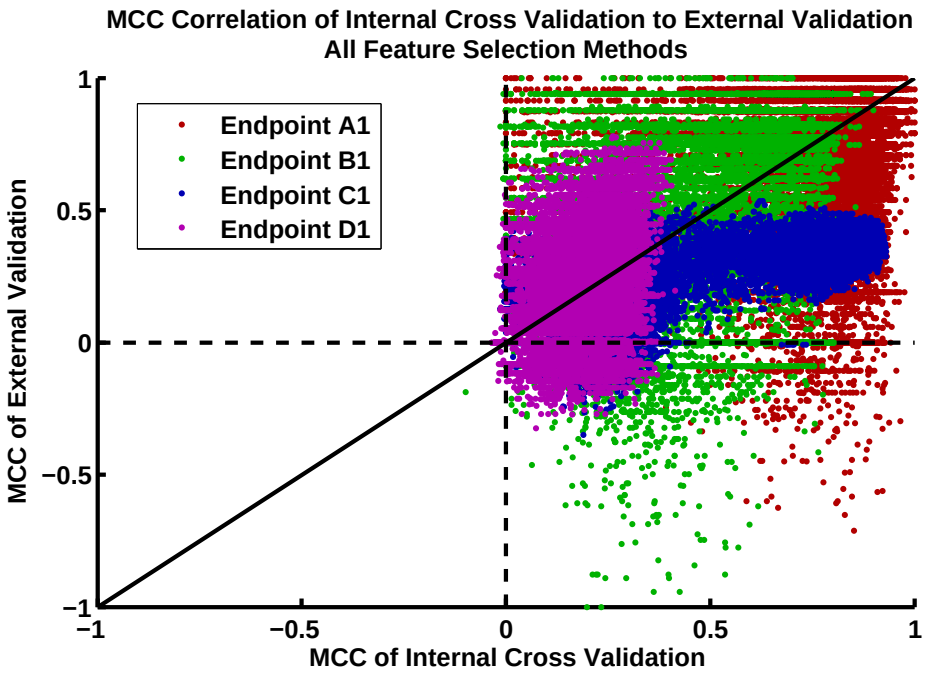
due to the larger sample sizes and relatively smaller batch effect. Nevertheless, validation performance is still highly variable, again, due to the variety of feature selection methods. The predictive performance assessed using the MCC, AUC, and accuracy all show similar behavior.

Swapping the training and testing data for each clinical endpoint reveals the extent of the batch effect within the A2 and B2 renal cancer datasets (**Figure 40**). Recall that this batch effect results in a large number of highly differentially expressed genes from feature selection. Here, we see that cross validation is severely optimistically biased while the variance of estimated model performance remains similar to that of the original data in **Figure 39**. The prostate cancer and breast cancer endpoints for the swapped analysis still have relatively tighter distributions. The prostate cancer endpoints appear to be less subject to over-fitting in the swapped analysis compared to the original analysis.

5.3.3 Modeling Factors Affecting Performance

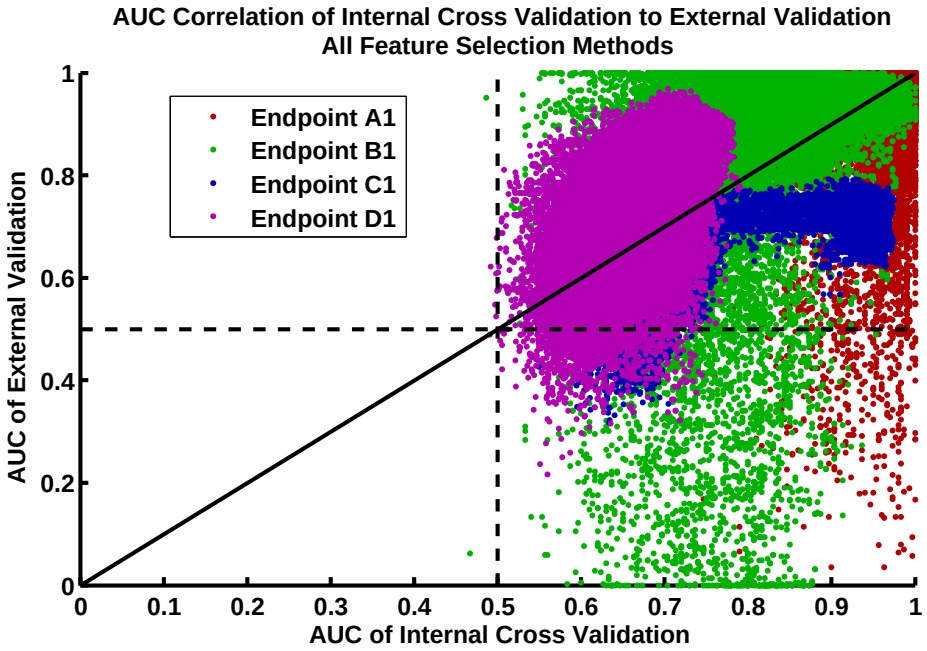
We are interested in identifying modeling factors that most contribute to variance in prediction performance. For each clinical endpoint, we perform an n-way fixed-effects analysis of variance (ANOVA). We consider three modeling factors: classifier, feature size, and feature selection methods. Feature sizes include 5, 10, 15, 20, 25, and 30 (selected as the top N genes after ranking). Feature selection methods include 258 (or 211 for the D1 and D2 endpoints) filter and wrapper-based methods. Classifiers include 30 linear and kernel based methods. These factors are listed in **Table 21**. We isolate performance metrics (MCC and AUC) as well as internal cross validation, external (blind) validation, and the absolute difference between internal and external validation.

The type of classifier appears to contribute the largest percentage of variance for the A1 and A2 endpoints when considering MCC as the performance metric (renal

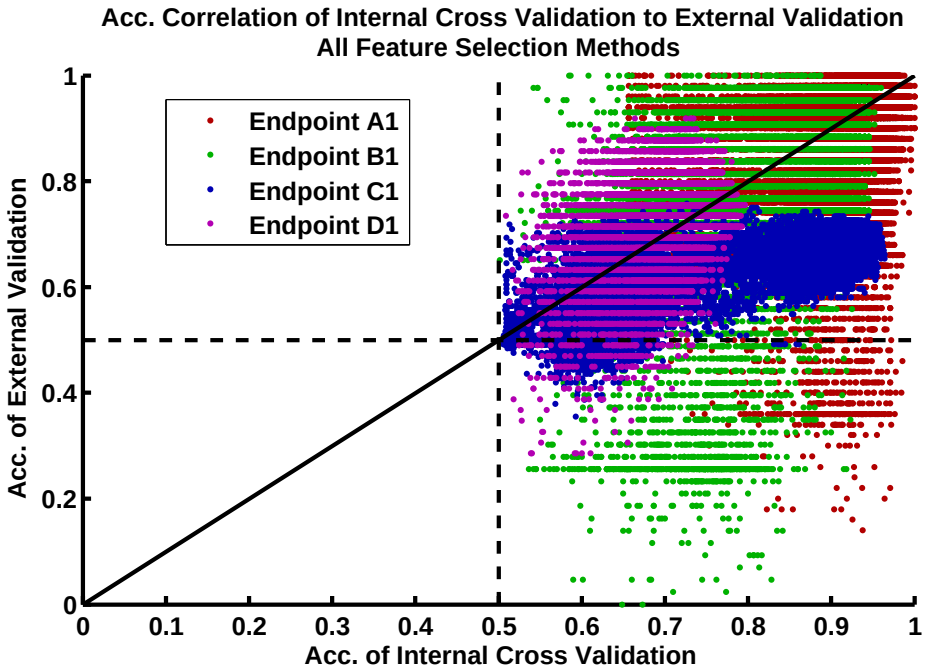


(a) Predictive model performance correlation between internal cross validation and external validation using the MCC metric.

Figure 39: Predictive model performance correlation between internal cross validation and external validation for the A1, B1, C1, and D1 endpoints. When considering all models, correlation is generally very poor for the A1 and B1 endpoints (renal cancer) due to small sample size as well as batch effect. Validation performance of the C1 endpoint (prostate cancer) is not as variable as the other endpoints. However, cross validation seems to overestimate predictive performance for this endpoint. Results are similar for each of the three performance metrics: MCC (**39(a)**), AUC (**39(b)**, next page), and Accuracy (**39(c)**, next page).

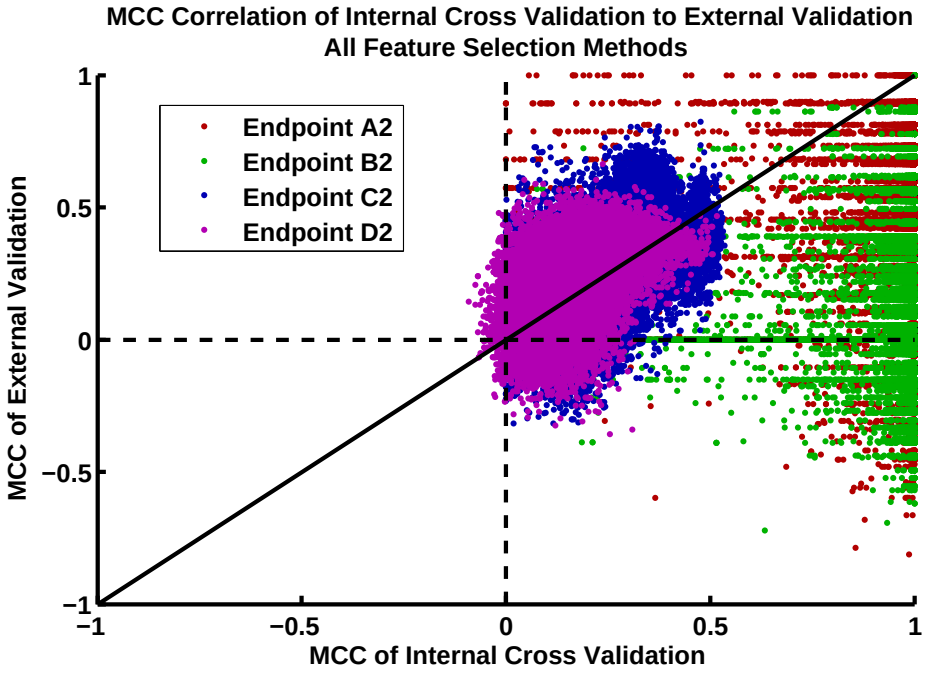


(b) Predictive model performance correlation between internal cross validation and external validation using the AUC metric.



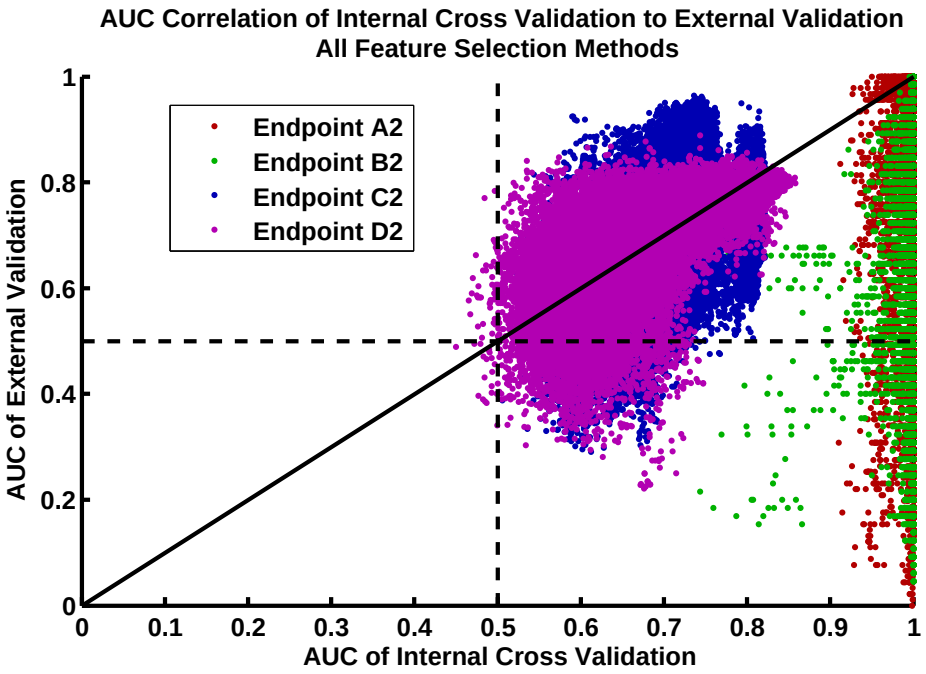
(c) Predictive model performance correlation between internal cross validation and external validation using the accuracy metric.

Figure 39 parts (b) and (c). Figure part (a) and full caption on the previous page.

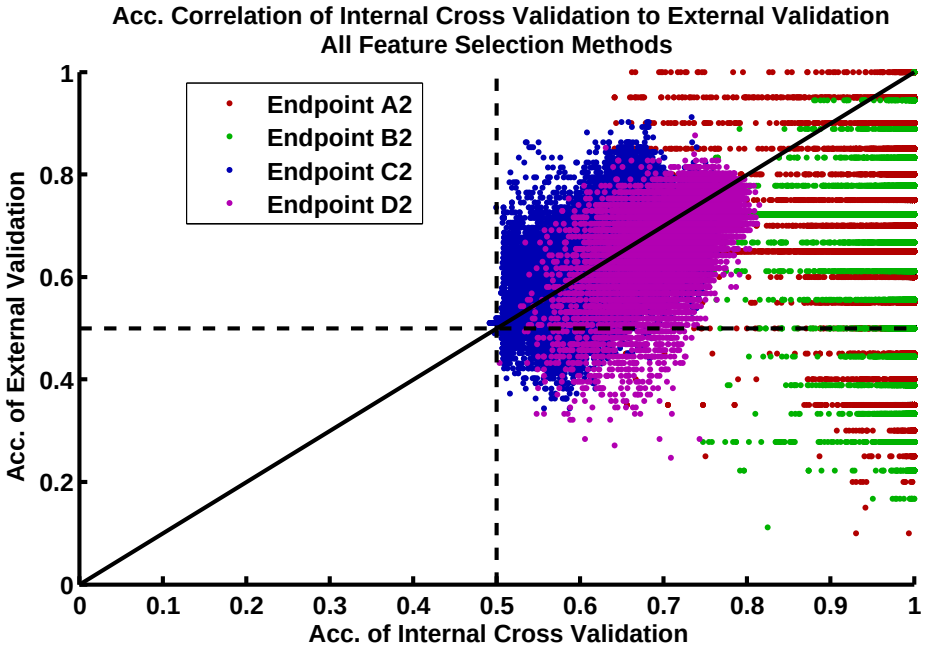


(a) Predictive model performance correlation between internal cross validation and external validation using the MCC metric, swapped training and testing data.

Figure 40: Predictive model performance correlation between internal cross validation and external validation for the A2, B2, C2, and D2 endpoints. When considering all models, correlation is generally very poor for the A2 and B2 endpoints (renal cancer) due to small sample size as well as batch effect. The batch effect within the A2 and B2 datasets explain the optimistic estimation of cross validation. Results are similar for each of the three performance metrics: MCC (**40(a)**), AUC (**40(b)**, next page), and Accuracy (**40(c)**, next page).



(b) Predictive model performance correlation between internal cross validation and external validation using the AUC metric, swapped training and testing data.



(c) Predictive model performance correlation between internal cross validation and external validation using the accuracy metric, swapped training and testing data.

Figure 40 parts (b) and (c). Figure part (a) and full caption on the previous page.

cancer, **Figure 41**). The AUC metric, however, seems to be most affected by feature size. This result is consistent for internal validation, external validation, and the absolute difference between internal and external validation. A possible explanation is that MCC is more sensitive to classifier parameters while AUC is not.

The variance of the MCC metric is also significantly affected by classifier type for the B1 endpoint (**Figure 42**, top left). In this case however, the effect of feature size on AUC prediction performance is not so clear (**Figure 42**, top right). Feature size contributes most to the AUC variability of cross validation and external validation for the B1 endpoint, but is not the predominant contributing factor to the variance of the difference between CV and EV.

For the prostate cancer data (**Figure 43**, top right), we see that the feature selection method is the largest contributing factor to AUC performance variance. Overall, in all endpoints, MCC variance is most affected by the choice of classifier, whereas the source of variability in the AUC metric depends on the clinical endpoint.

5.3.4 Biological Knowledge Improves Clinical Prediction Performance

In this section, we isolate the variability due to the feature selection method and determine if any relationship exists between the biological relevance of feature selection and prediction performance. We begin by removing the majority of feature selection methods from the population of models for each endpoint and observe any changes in cross validation and validation performance (**Figure 45**, **Figure 46**, **Figure 47**, and **Figure 48**). In **Figure 45**, we observe immediately that reducing the feature selection methods to only include the top 10 according to biological relevance removes a majority of the models that perform poorly for external validation. Endpoints affected most significantly are the renal cancer data, A1 and B1. We also see that models for the prostate cancer endpoint (C1) reduce to a set that tends to over-fit

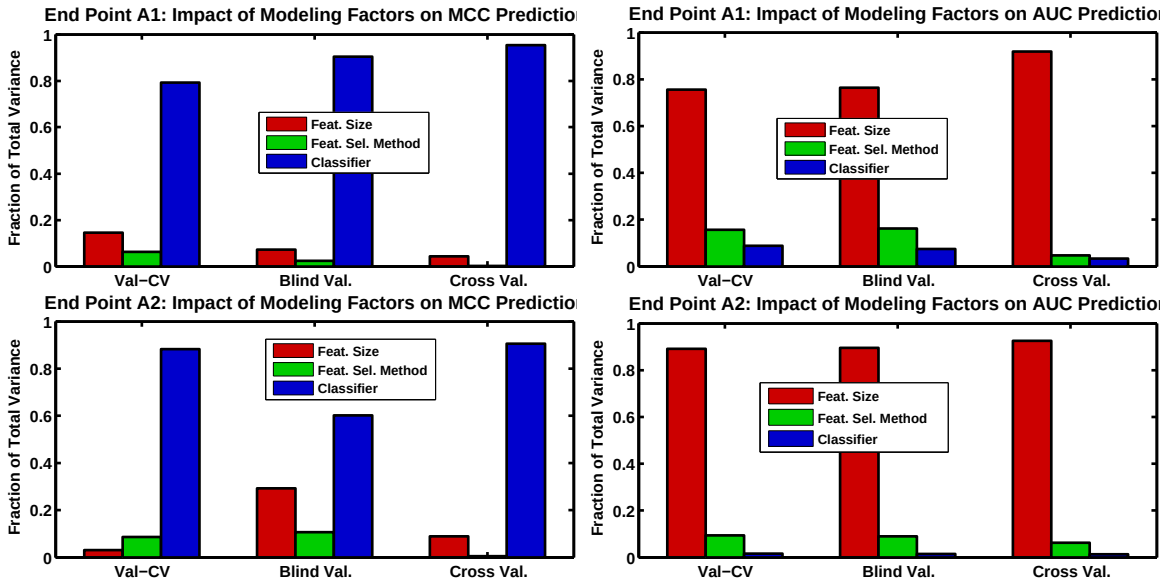


Figure 41: Factors that affect internal cross validation, external validation, and the difference between internal and external validation for renal cancer endpoints A1 and A2 comparing the CC and ONC/CHR subtypes. The MCC performance metric seems to be affected by classifier modeling factors, including the kernel and associated parameters. When using the AUC performance metric, feature size predominantly affects the variability of performance. The effect of feature selection method, though small, seems to be larger on the blind validation compared to internal cross validation.

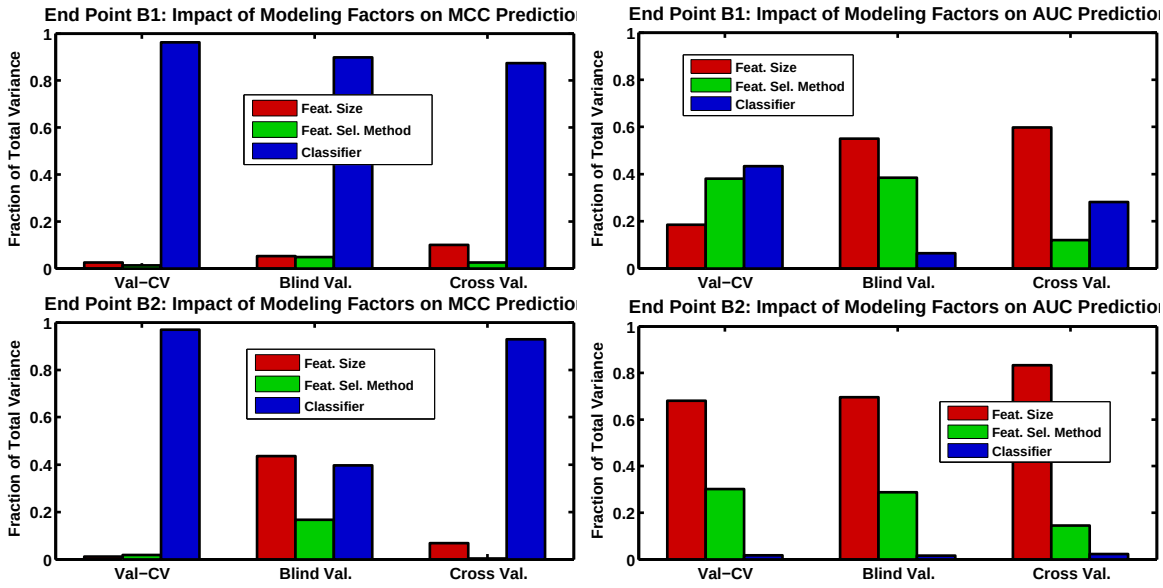


Figure 42: Factors that affect internal cross validation, external validation, and the difference between internal and external validation for renal cancer endpoints B1 and B2 comparing the CC and PAP subtypes. The MCC performance metric seems to be affected by classifier modeling factors, including the kernel and associated parameters. When using the AUC performance metric, feature size predominantly affects the variability of performance. The effect of feature selection method, though small, seems to be larger on the blind validation compared to internal cross validation.

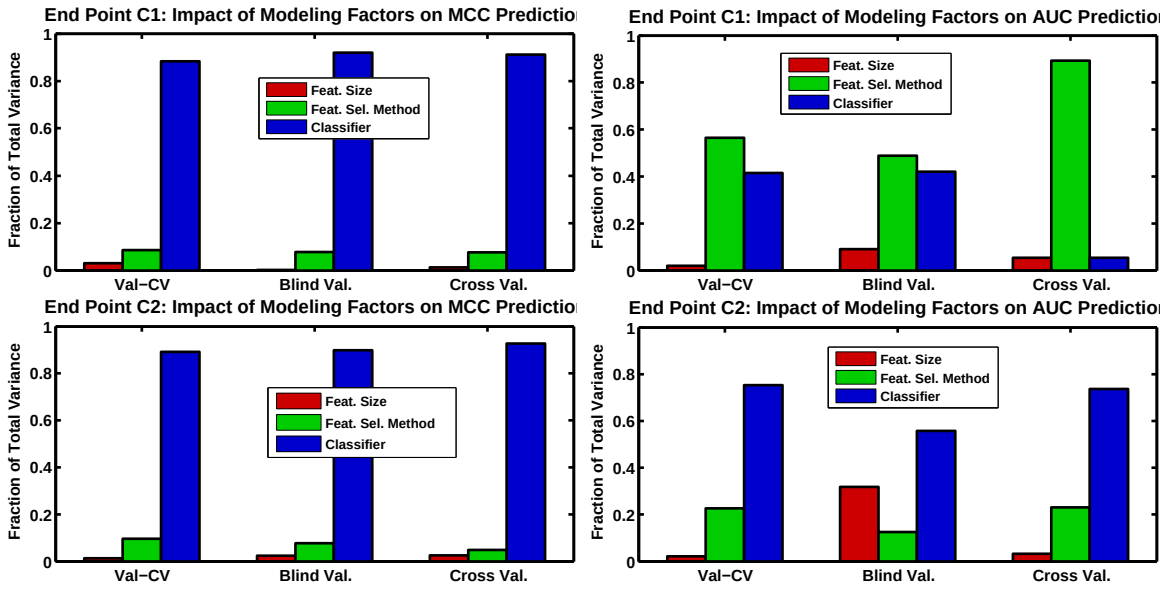


Figure 43: Factors that affect internal cross validation, external validation, and the difference between internal and external validation for prostate cancer endpoints C1 and C2 comparing tumor and normal adjacent tissue. The MCC performance metric seems to be affected by classifier modeling factors, including the kernel and associated parameters. When using the AUC performance metric, feature size predominantly affects the variability of performance. The effect of feature selection method, though small, seems to be larger on the blind validation compared to internal cross validation.

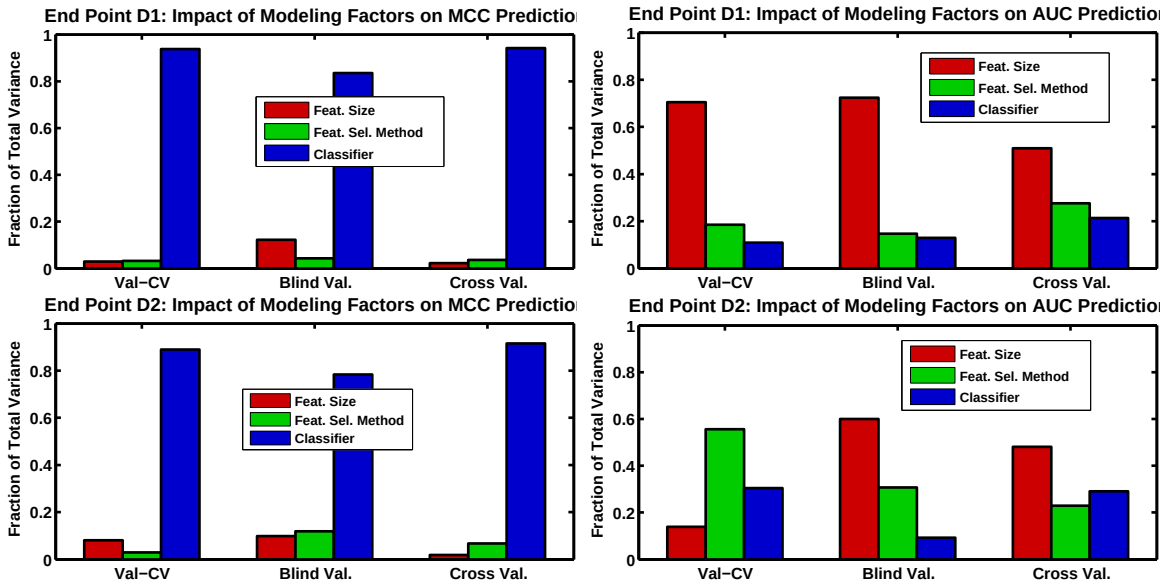


Figure 44: Factors that affect internal cross validation, external validation, and the difference between internal and external validation for breast cancer endpoints D1 and D2 comparing treatment outcomes. The MCC performance metric seems to be affected by classifier modeling factors, including the kernel and associated parameters. When using the AUC performance metric, feature size predominantly affects the variability of performance. The effect of feature selection method, though small, seems to be larger on the blind validation compared to internal cross validation.

during cross validation. We may interpret this as an increase in cross-validation performance as biological relevance increases. This is a natural interpretation, since the biological relevance is measured from the training data. However, this is not the case for the A1, B1, and D1 datasets, each of which results in only slight increases in cross validation performance as a whole. The majority of the performance improvements for A1 and B1 occur in the validation performance. When we further reduce the space of feature selection methods to only the top method, the distributions of model performances tightens further (**Figure 47** and **Figure 48**). Of particular note is the increase in validation performances of A1, B1, and C1 such that the cross validation performance appears to under-estimate validation performance for the MCC, AUC, and accuracy metrics (except for endpoint A1 for the accuracy performance metric).

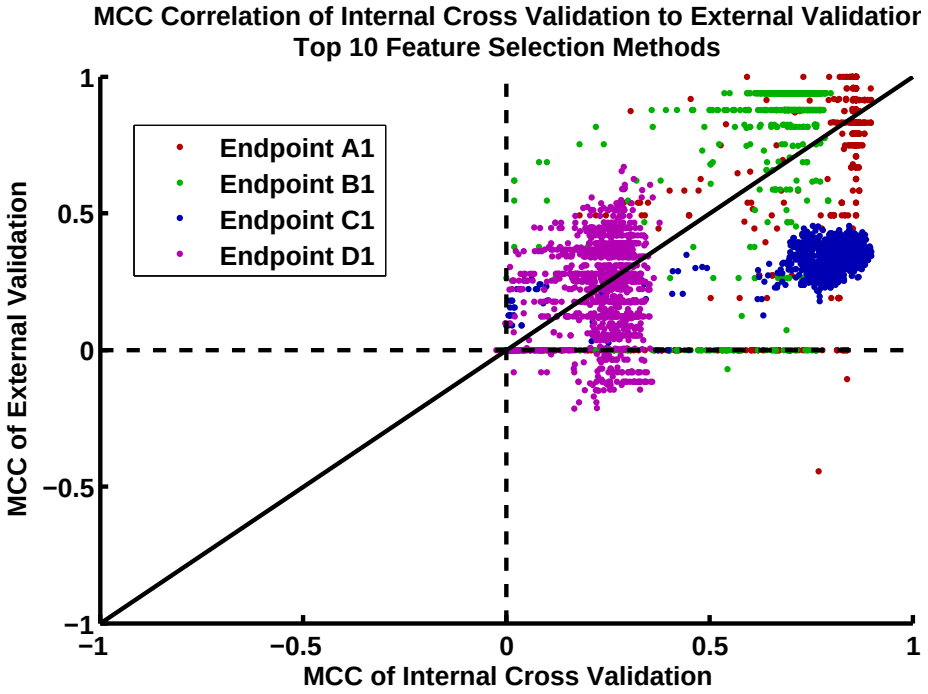
The effect of reducing the feature selection model space is similar after swapping the training and testing data (**Figure 46**). Again, due to the batch effect within the A2 and B2 renal cancer datasets, cross validation tends to over-estimate performance. However, we see that reduction of the feature selection methods to the top 10 in terms of biological relevance nominally improves A2 and B2 validation performance and even more so for the top feature selection method (**Figure 48**). As we remove feature selection methods, the concordance between the cross validation estimate and external validation for the prostate and breast cancer endpoints C2 and D2 improves. In this case, there is no over-estimation of prostate cancer validation during cross validation. Overall, we see that using more biologically relevant feature selection methods tends to improve concordance between internal cross validation and external validation. Additionally, biologically relevant feature selection methods seem to remove a majority of the poorly performing models in terms of external validation.

A direct examination of external blind validation performance as a function of the biological relevance of feature selection is more revealing (**Figure 49**, **Figure 50**, **Figure 51**, and **Figure 52**). We compute the biological relevance using **Equation**

20 with the biological knowledge specifically identified for each clinical endpoint (**Table 16**, **Table 17**, **Table 18**, and **Table 19**). The 46440 (37980 for endpoints D1 and D2) prediction models for each clinical endpoint can be divided into 258 feature selection methods (211 for endpoints D1 and D2), resulting in 180 models (30 classifiers x 6 feature sizes). For each feature selection method, we compute the average external validation performance of the 180 models and plot these as a function of biological relevance. In all cases, external validation performance is positively correlated with biological relevance with statistical significance (p ¡ 0.05). Each of the clinical endpoints has a varying degree of correlation. Although endpoints A2 and B2 severely overestimate predictor performance due to an internal batch effect, these models still show a slight improvement as biological relevance of the feature selection method increases.

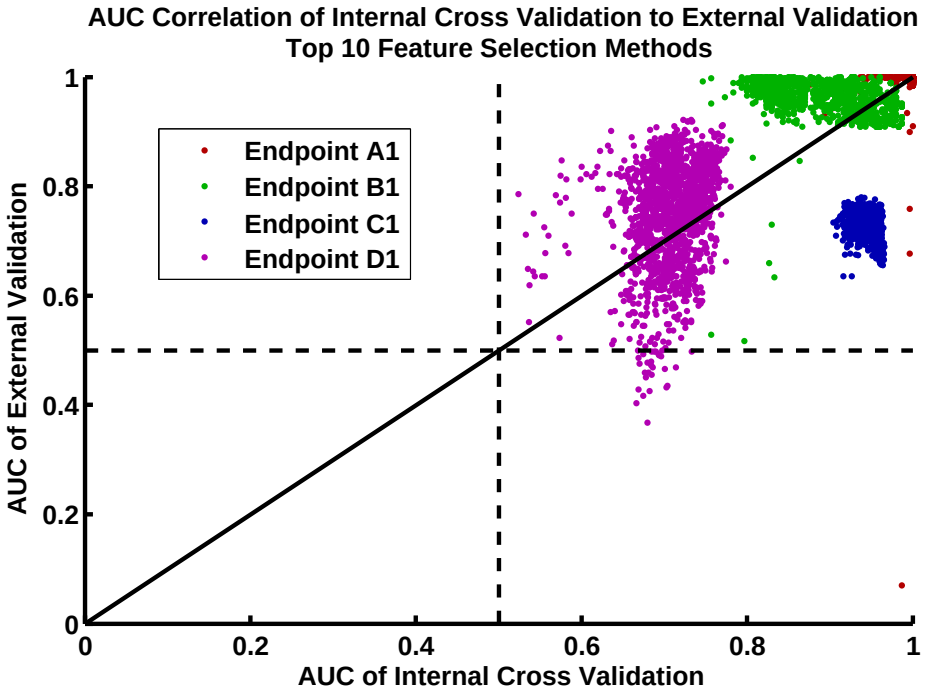
5.4 Conclusion

We have performed a systematic study of the feasibility of predictive biomarkers using four clinical endpoints. Furthermore, we have shown that using biologically relevant feature selection methods reduces the variability of predictive performance estimates and improves overall predictive performance on blind external validation data. The clinical endpoints in this study are diverse and include renal cancer, prostate cancer, and breast cancer. Although the incidence of renal cancer is less than that of either prostate or breast cancer, it is still a significant clinical problem due to the heterogeneity of the disease. Renal cancer includes several subtypes that are difficult to distinguish, yet require different treatment regimens [124]. Thus, clinical biomarkers that are able to identify particular subtypes will enable physicians to improve the success of treatment. In contrast, prostate cancer is the most common form of cancer in men and early detection usually results in successful treatment. However, treatment decisions may not be optimal. For example, some patients may require a

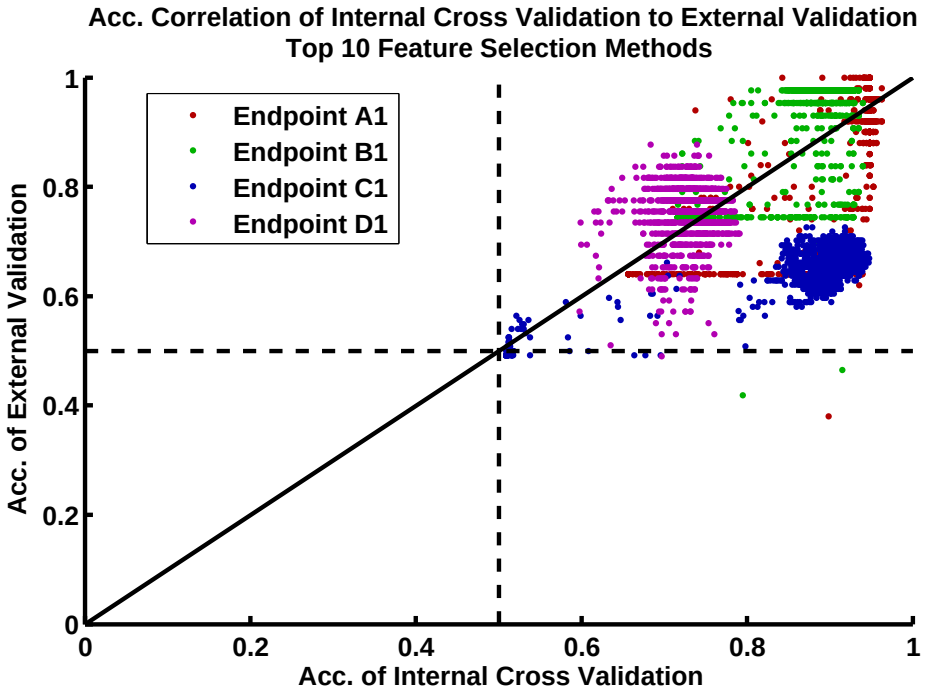


(a) Predictive model performance correlation between internal cross validation and external validation using the MCC metric, top 10 biologically relevant feature selection methods.

Figure 45: Predictive model performance correlation between internal cross validation and external validation for the A1, B1, C1, and D1 endpoints. As the models are filtered to include only the top 10 biologically relevant feature selection methods, external validation performance improves. Cross validation of predictive models using samples from the A1, B1, and D1 datasets generally predict classification performance on the A2, B2, and D2 data samples, respectively. Cross validation of models created with C1 data samples tend to over-estimate prediction performance using C2 samples. Results are similar for each of the three performance metrics: MCC (**45(a)**), AUC (**45(b)**, next page), and Accuracy (**45(c)**, next page).

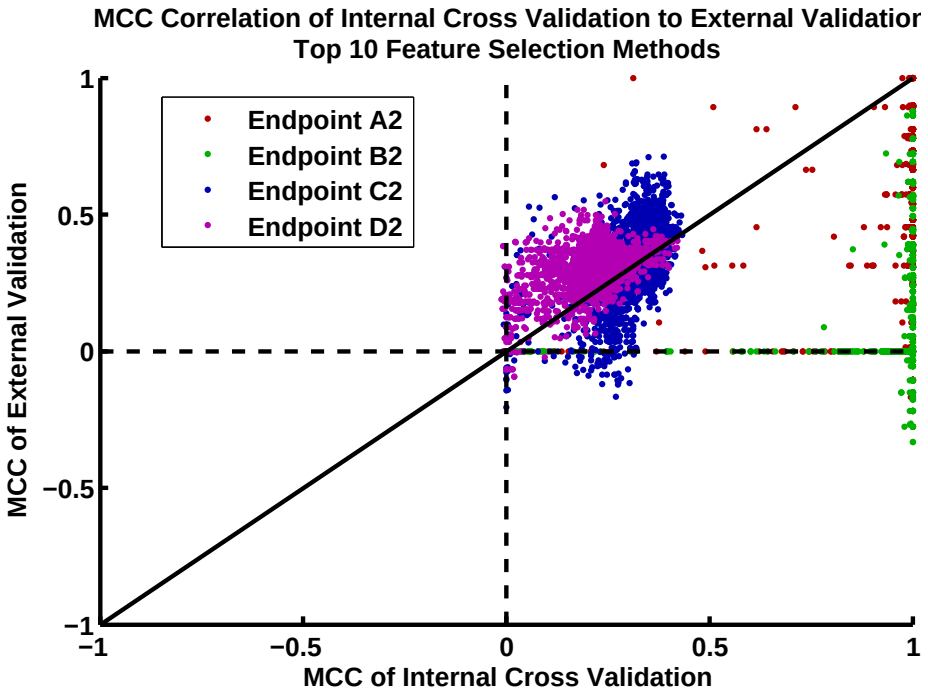


(b) Predictive model performance correlation between internal cross validation and external validation using the AUC metric, top 10 biologically relevant feature selection methods.



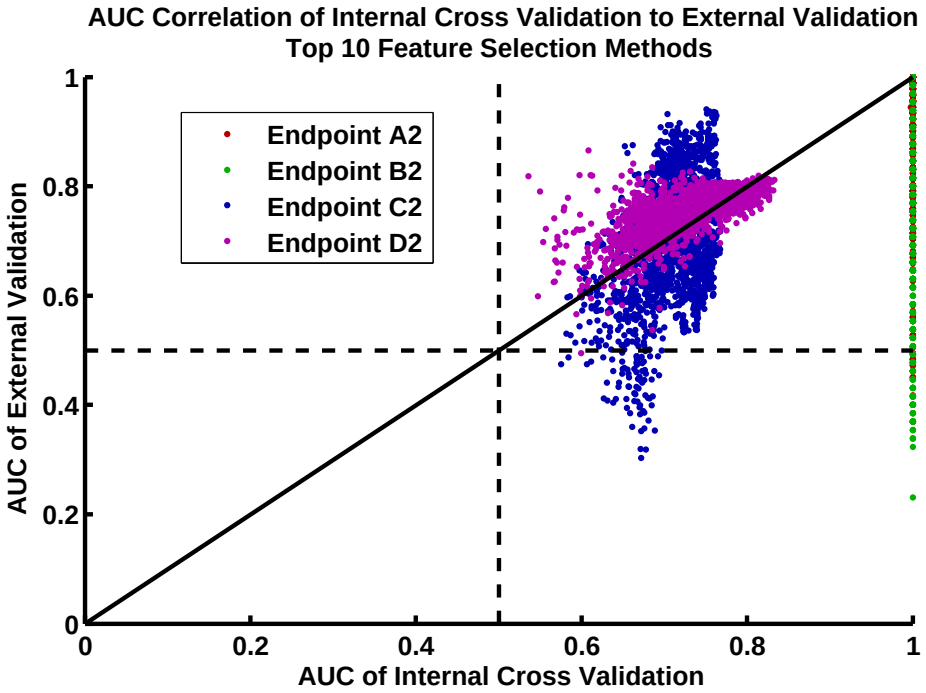
(c) Predictive model performance correlation between internal cross validation and external validation using the accuracy metric, top 10 biologically relevant feature selection methods.

Figure 45 parts (b) and (c). Figure part (a) and full caption on the previous page.

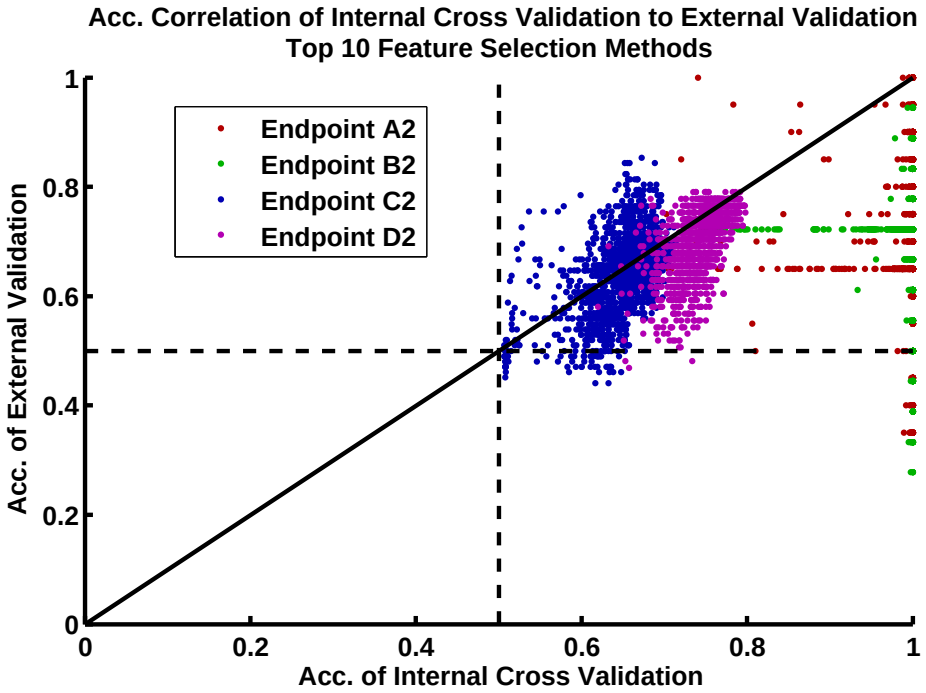


(a) Predictive model performance correlation between internal cross validation and external validation (swapped) using the MCC metric, top 10 biologically relevant feature selection methods.

Figure 46: Predictive model performance correlation between internal cross validation and external validation for the A2, B2, C2, and D2 endpoints. When considering all models, correlation is generally very poor for the A2 and B2 endpoints due to small sample size as well as batch effect. The batch effect between classes within the A2 and B2 endpoints is severe, leading to over-estimation of prediction performance. Despite this, we see that as the models are filtered to include the top 10 biologically relevant feature selection methods, external validation performance improves. Cross validation of predictive models using samples from the C2 and D2 datasets generally predict classification performance on the C1 and D1 data samples, respectively. Results are similar for each of the three performance metrics: MCC (**46(a)**), AUC (**46(b)**, next page), and Accuracy (**46(c)**, next page).

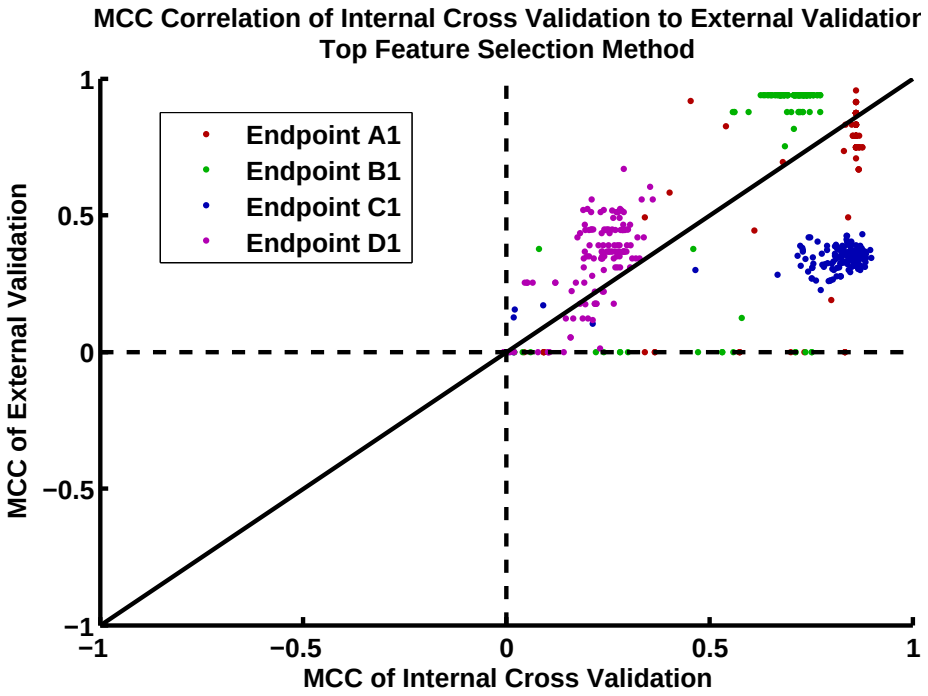


(b) Predictive model performance correlation between internal cross validation and external validation (swapped) using the AUC metric, top 10 biologically relevant feature selection methods.



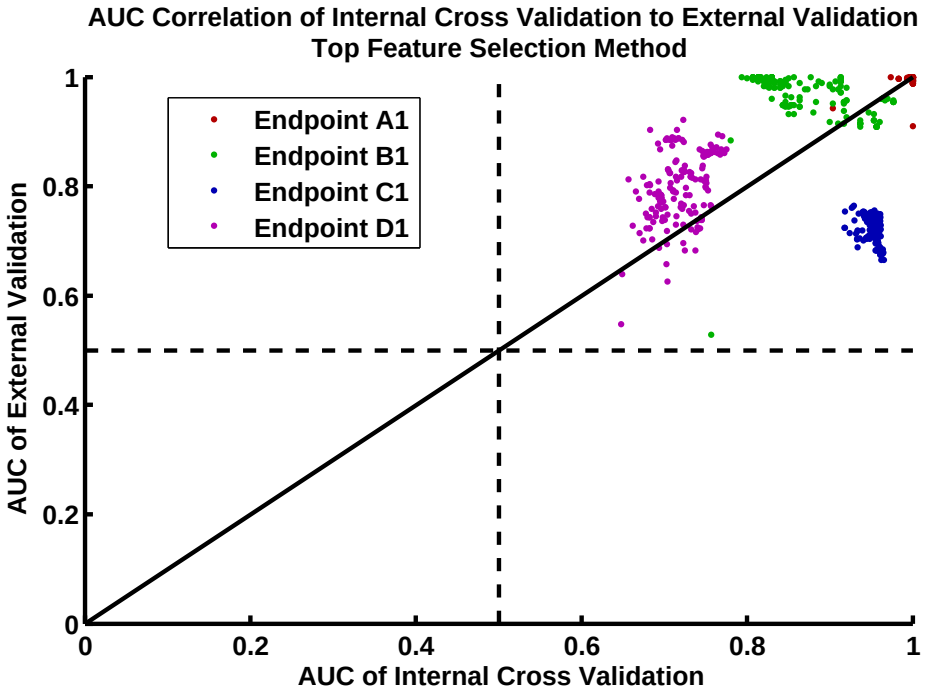
(c) Predictive model performance correlation between internal cross validation and external validation (swapped) using the accuracy metric, top 10 biologically relevant feature selection methods.

Figure 46 parts (b) and (c). Figure part (a) and full caption on the previous page.

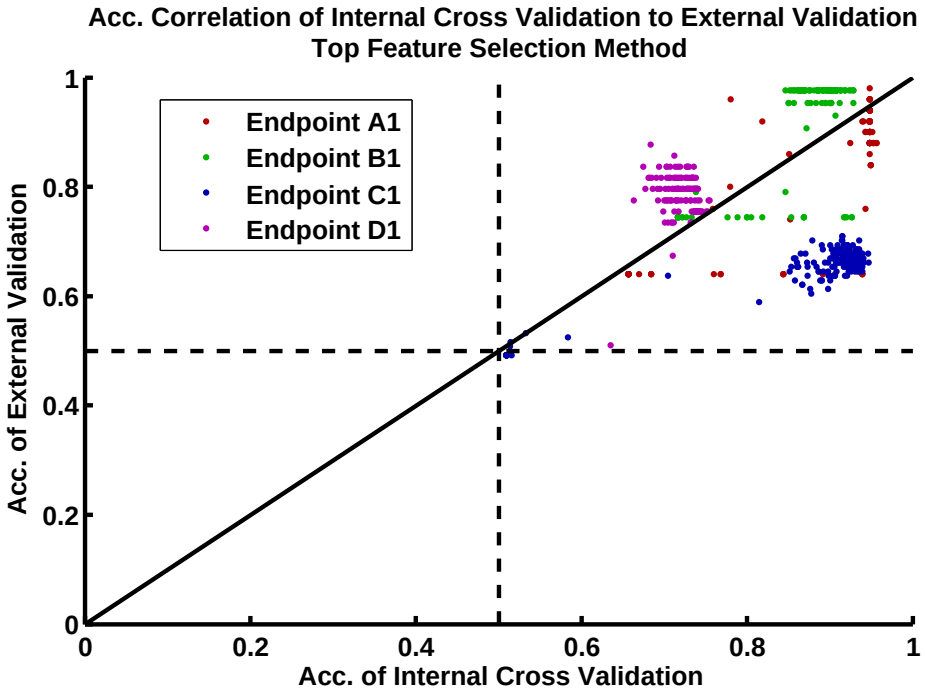


(a) Predictive model performance correlation between internal cross validation and external validation using the MCC metric and the most biologically relevant feature selection method.

Figure 47: Predictive model performance correlation between internal cross validation and external validation for the A1, B1, C1, and D1 endpoints. As the models are filtered to include only the most biologically relevant feature selection method, external validation performance improves even further. Cross validation of predictive models using samples from the A1, B1, and D1 datasets generally predict classification performance on the A2, B2, and D2 data samples, respectively. Cross validation of models created with C1 data samples tend to over-estimate prediction performance using C2 samples. Results are similar for each of the three performance metrics: MCC (**47(a)**), AUC (**47(b)**, next page), and Accuracy (**47(c)**, next page).

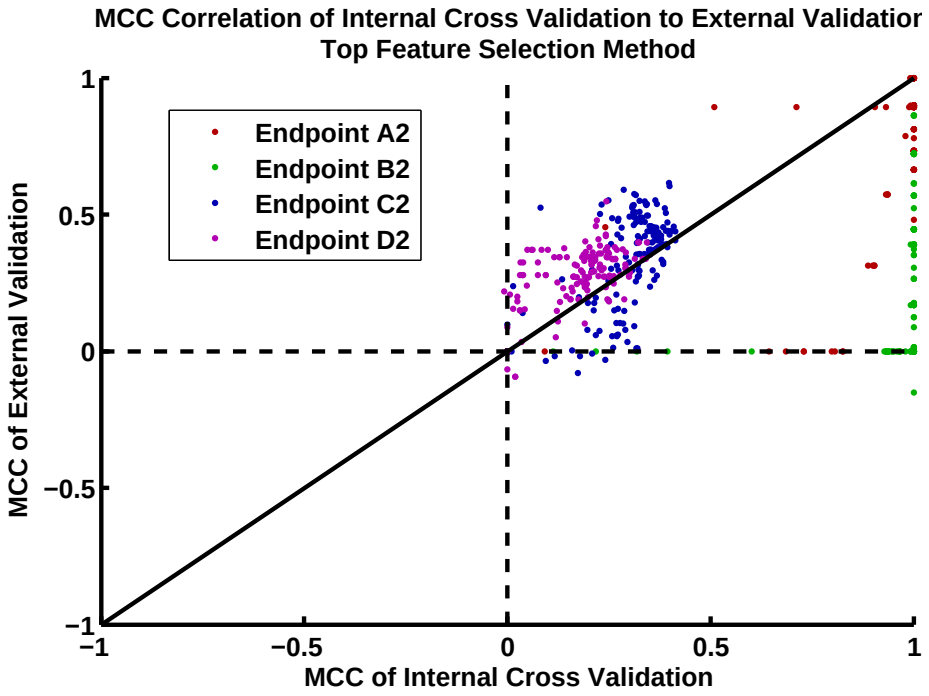


(b) Predictive model performance correlation between internal cross validation and external validation using the AUC metric and the most biologically relevant feature selection method.



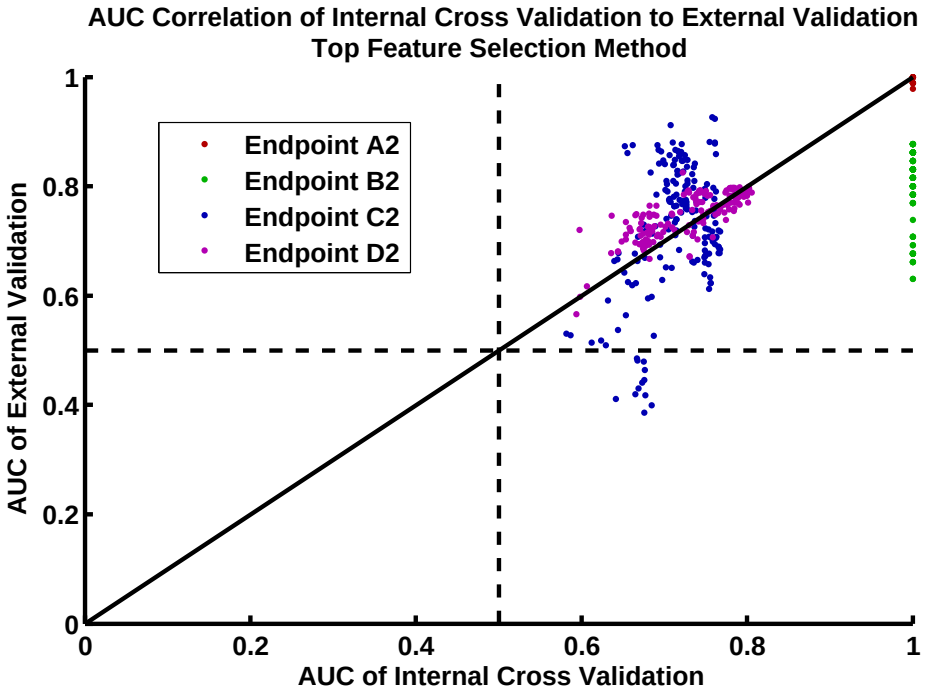
(c) Predictive model performance correlation between internal cross validation and external validation using the accuracy metric and the most biologically relevant feature selection method.

Figure 47 parts (b) and (c). Figure part (a) and full caption on the previous page.

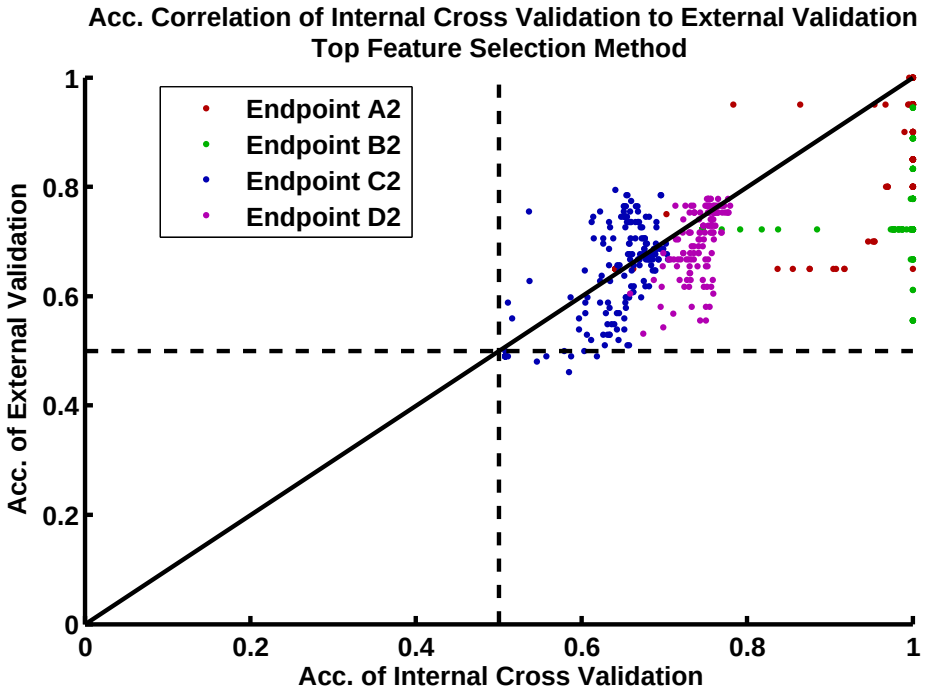


(a) Predictive model performance correlation between internal cross validation and external validation (swapped) using the MCC metric and the most biologically relevant feature selection method.

Figure 48: Predictive model performance correlation between internal cross validation and external validation for the A2, B2, C2, and D2 endpoints. When considering all models, correlation is generally very poor for the A2 and B2 endpoints due to small sample size as well as batch effect. The batch effect between classes within the A2 and B2 endpoints is severe, leading to over-estimation of prediction performance. Despite this, we see that as the models are filtered to include only the most biologically relevant feature selection method, external validation performance improves. Cross validation of predictive models using samples from the C2 and D2 datasets generally predict classification performance on the C1 and D1 data samples, respectively. Results are similar for each of the three performance metrics: MCC (**48(a)**), AUC (**48(b)**, next page), and Accuracy (**48(c)**, next page).



(b) Predictive model performance correlation between internal cross validation and external validation (swapped) using the AUC metric and the most biologically relevant feature selection method.



(c) Predictive model performance correlation between internal cross validation and external validation (swapped) using the accuracy metric and the most biologically relevant feature selection method.

Figure 48 parts (b) and (c). Figure part (a) and full caption on the previous page.

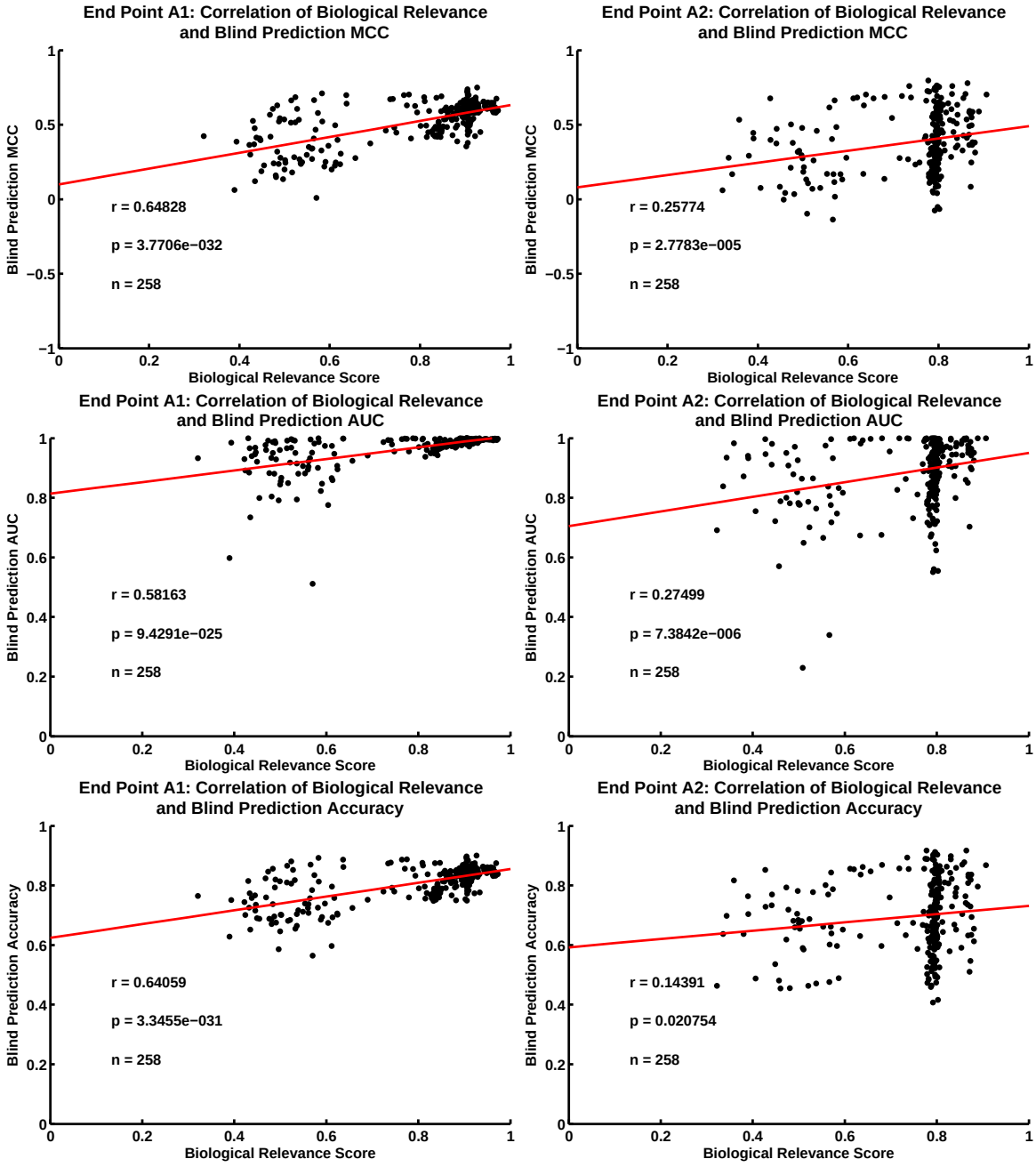


Figure 49: Biological relevance of feature selection improves predictive model performance on blind renal cancer data comparing CC and ONC/CHR subtypes. The correlation between external prediction performance and biological relevance is stronger when models are trained using the A1 dataset (left column). Although models trained using the A2 dataset (right column) tend to over-fit, there is still a positive correlation between biological relevance and predictive performance. Correlations are statistically significant ($p < 0.05$) for all performance metrics: MCC (top row), AUC (middle row), and Accuracy (bottom row).

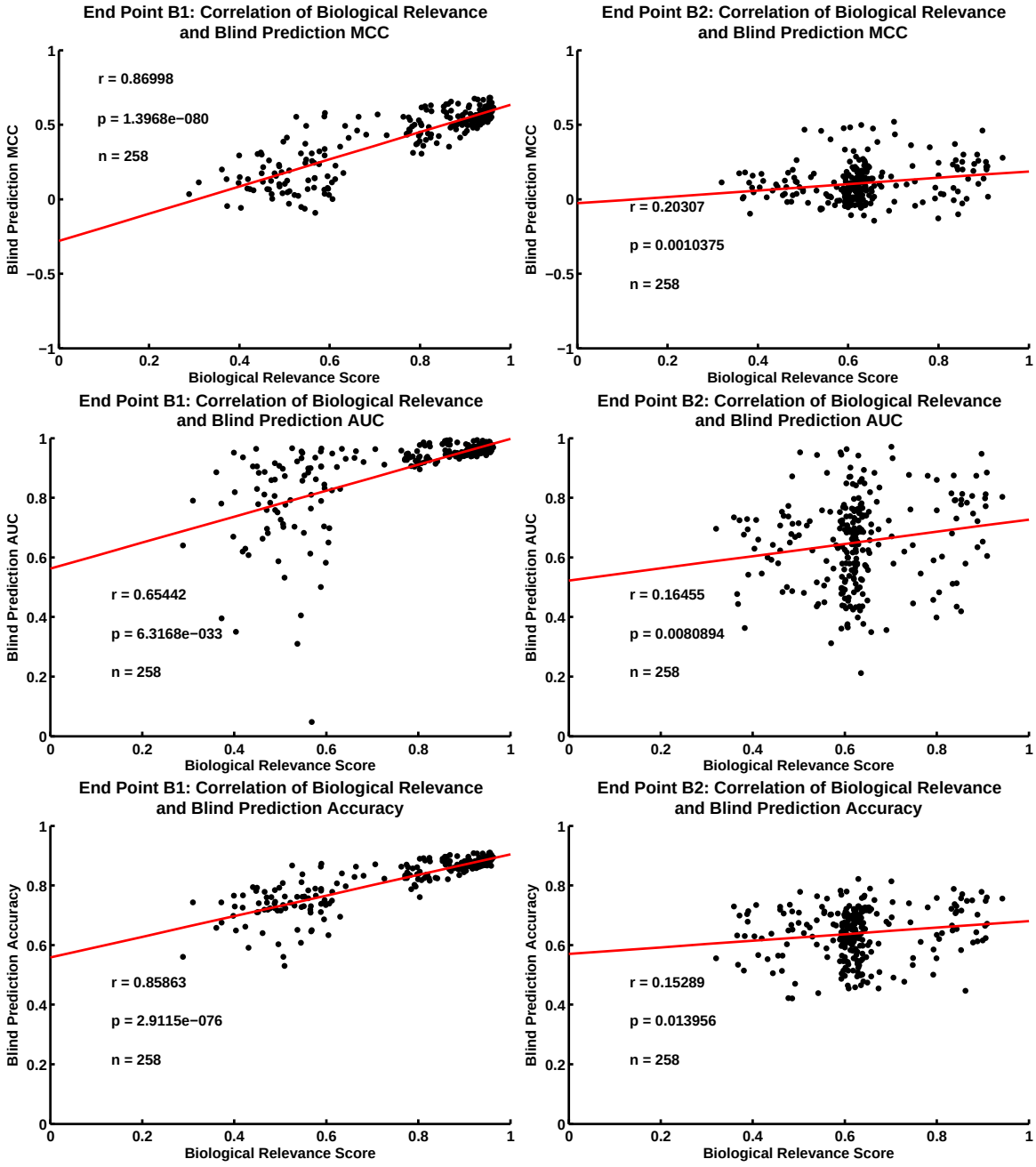


Figure 50: Biological relevance of feature selection improves predictive model performance on blind renal cancer data comparing CC and PAP subtypes. The correlation between external prediction performance and biological relevance is stronger when models are trained using the B1 dataset (left column). Although models trained using the B2 dataset (right column) tend to over-fit, there is still a positive correlation between biological relevance and predictive performance. Correlations are statistically significant ($p < 0.05$) for all performance metrics: MCC (top row), AUC (middle row), and Accuracy (bottom row).

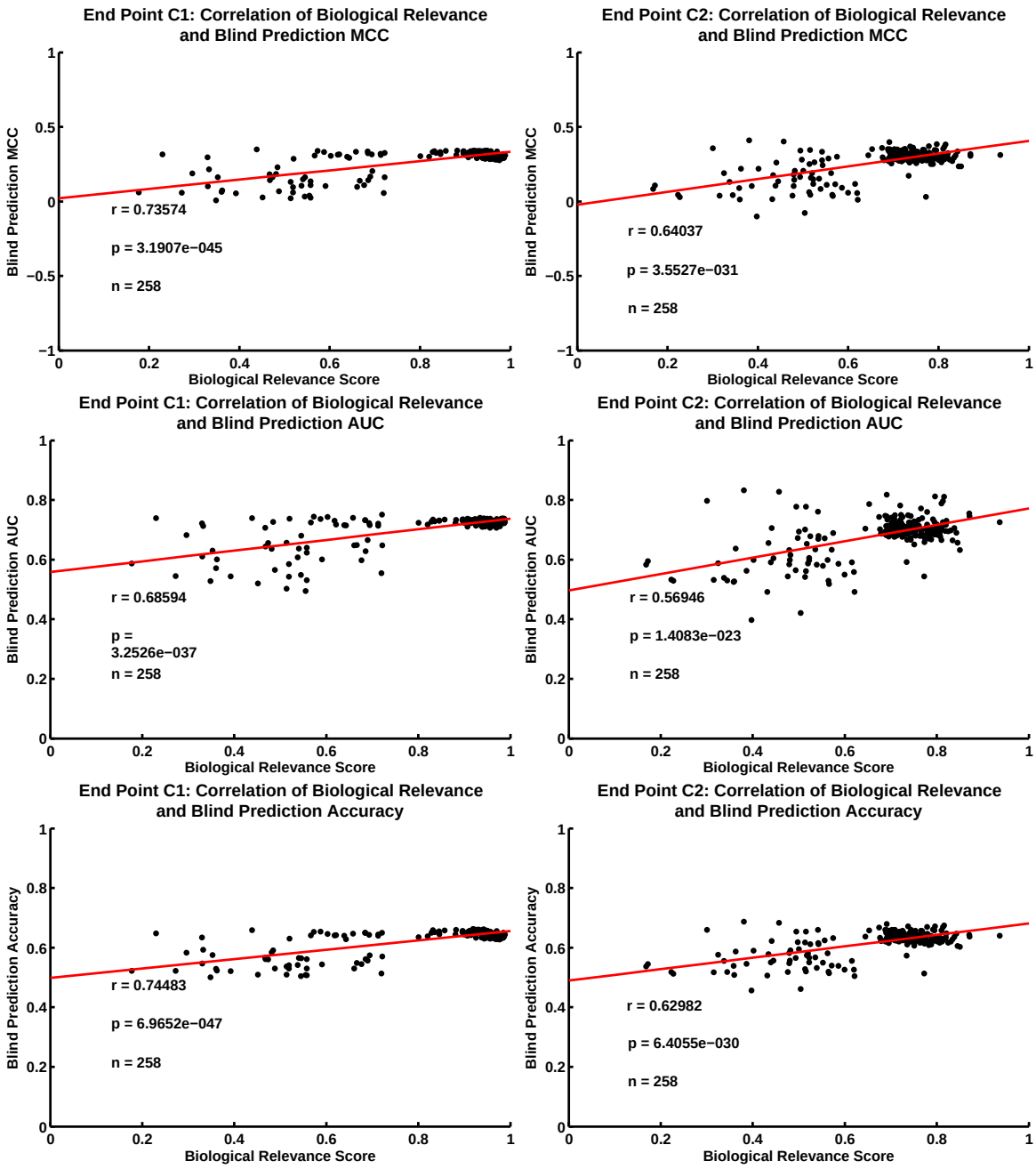


Figure 51: Biological relevance of feature selection improves predictive model performance on blind prostate cancer data comparing tumor and adjacent normal tissue. There is a positive correlation between biological relevance and predictive performance regardless of the training data, C1 (left column) or C2 (right column). Correlations are statistically significant ($p < 0.05$) for all performance metrics: MCC (top row), AUC (middle row), and Accuracy (bottom row).

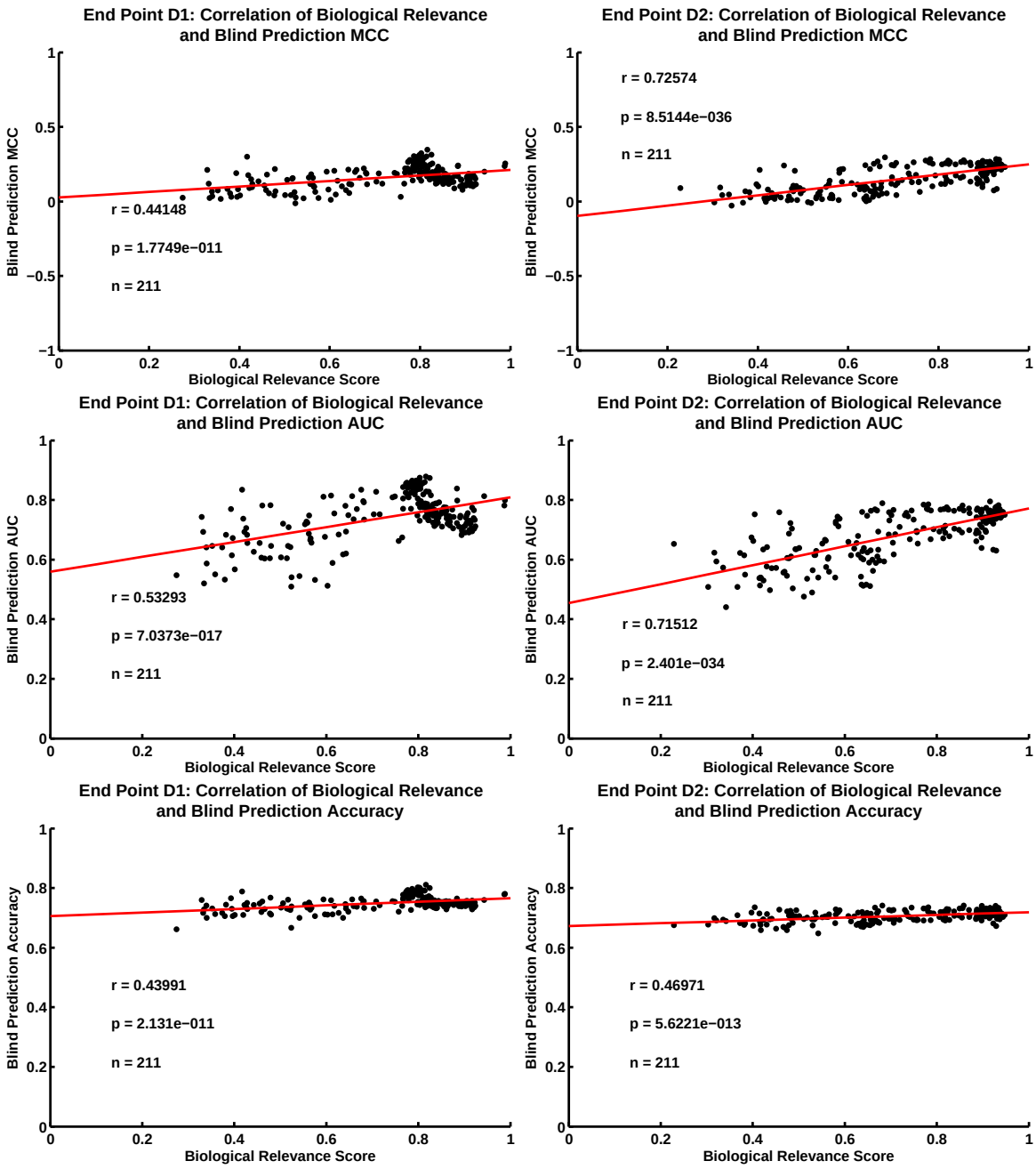


Figure 52: Biological relevance of feature selection improves predictive model performance on blind breast cancer data comparing treatment outcomes. There is a positive correlation between biological relevance and predictive performance regardless of the training data, D1 (left column) or D2 (right column). Correlations are statistically significant ($p < 0.05$) for all performance metrics: MCC (top row), AUC (middle row), and Accuracy (bottom row).

prostatectomy while others may only require radiation treatment. The decision to use one method or the other may be arbitrary, resulting in sub-optimal treatment. Many studies have attempted to identify biomarkers that can predict the success of specific treatment options. Here, we are interested in biomarkers that can distinguish between prostate cancer tissue and normal tissue adjacent to the cancer tissue. Such biomarkers may help to improve the accuracy of prostate cancer diagnosis from tissue biopsies. Breast cancer treatment suffers from similar problems. Due to a wide variety of treatment options, there are many studies that attempt to identify biomarkers that predict specific treatment outcomes. Here, we examine predictors that can identify patients that are likely to respond well to a specific treatment regimen of chemotherapy [53]. In general, the success of clinical predictors derived from high-throughput data would greatly improve clinical applications.

The validation performance of our predictors varies widely within each endpoint, especially for the renal cancer datasets. This variability may be attributed to dataset-specific factors, such as sample size and batch effect. The renal cancer datasets were small compared to the prostate and breast cancer data, reflecting the relative differences in disease incidence. Batch effect within datasets and between training and testing datasets results in over-fitting, in which the predictive model becomes specialized to sub-population of patients and cannot generalize to the population as a whole.

Overall, we have seen that the success of a clinical predictor depends on the feature selection method. We considered a large variety of feature selection methods, many of which were found to be biologically irrelevant according to previously validated biomarkers. Removal of these biologically irrelevant ranking methods improved overall validation performance as well as concordance between cross validation estimates and external validation. Moreover, we found that the biological relevance of a feature selection method is positively correlated with blind validation performance.

CHAPTER VI

OMNIBIOMARKER: A TRANSLATIONAL BIOINFORMATICS APPLICATION

6.1 Introduction

omniBiomarker is a bioinformatics application that serves as both a microarray data repository as well as a tool for identifying clinically useful candidate biomarkers. Biomarker identification from high-throughput microarray data for clinical prediction is sensitive to analysis parameters [88]. As a result, candidate biomarker lists can be difficult to reproduce, limiting the efficiency of translating candidate biomarker lists to clinical applications. omniBiomarker addresses this problem by tuning steps in the analysis pipeline to a clinical problem based on prior biological knowledge [106]. **Figure 53** illustrates the data analysis pipeline that omniBiomarker encapsulates (left panel), with examples of system output (right panel). We can use clinically validated biomarkers as references with which to identify the most biologically relevant feature selection algorithms. Despite the lag between initial proposal of candidate biomarkers and final clinical validation, there are still a number of biomarkers that we can use as knowledge [156]. By integrating knowledge in this manner, we can overcome the curse of dimensionality problem and increase the reproducibility of clinical prediction. omniBiomarker also addresses the problem of community accessibility. It is developed with a focus on not only the novelty of the analysis pipeline, but also on the integration of these analytical steps into a user-friendly, web-accessible interface. This attribute of bioinformatics tools has become increasingly important as the gap between clinical applications and bioinformatics narrows. omniBiomarker is also caBIG-compatible, further increasing the interoperability of its functions with other

bioinformatics tools in the cancer research community [95].

Not surprisingly, many of these web-based applications implement functionality for several common steps in the data analysis pipeline. Despite the existence of many web-based tools for biomarker identification, we are still several steps removed from clinical applications. Before using these clinical biomarkers in clinical scenarios, we must interpret and verify their biological validity.

The availability of many software packages for each step in the biomarker identification pipeline enables us to choose from a variety of methods to suit our needs. However, the lack of an established data standard impedes our progress when we try to fit the pieces together [98]. For example, without translating the data format, we may not be able to use the data output of a quality control and normalization application in a subsequent clustering or feature selection application. Furthermore, we sometimes need to translate lists of gene symbols from a feature selection application before interpretation with a particular GO application [46]. The goals of these workflow applications support those of omniBiomarker by speeding up the process by which bioinformaticians can assess the clinical feasibility of a particular data-specific workflow. We can extend the philosophy of knowledge-driven algorithm selection to the more general knowledge-driven workflow selection. Thus, in order confidently choose a workflow we need to assess many different data analysis paths to determine the relevance of their results with respect to prior biological and clinical knowledge.

6.2 omniBiomarker: Web-Based Application

The primary aim of the omniBiomarker web-based application is to provide a microarray storage and analysis engine to the bioinformatics community. The unique functions of this application include 1) user-specific data privacy, 2) centralized gene expression data storage and manipulation, 3) a multitude of wrapper-based feature ranking algorithms, 4) centralized storage of analytical results as well as analytical

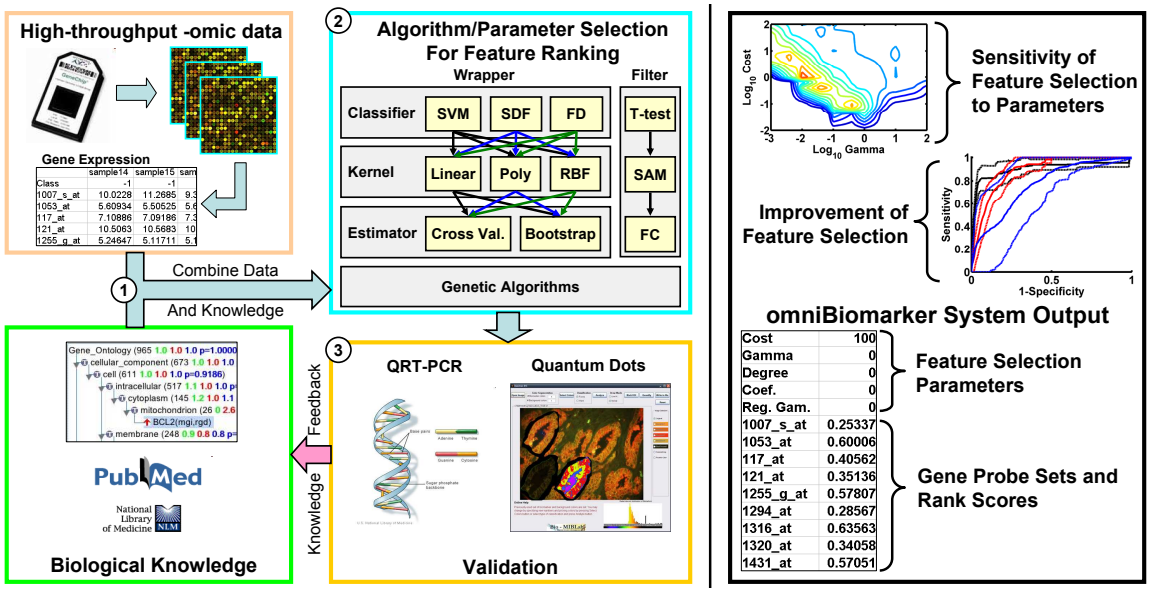


Figure 53: The knowledge-guided workflow for identifying differentially expressed genes consists of several steps (left panel). In step one, we collect high-throughput -omic data using technology such as microarrays as well as biological knowledge about the disease of interest. In step two, rather than identifying features using only data-driven methods, we use biological knowledge to guide the feature selection process and identify the algorithm that results in rankings with maximal biological relevance. Finally, in step three, we validate the candidate biomarkers with qRT-PCR or other imaging techniques in order to 1) ensure highly accurate clinical applications and 2) improve our biological knowledge. We have shown that feature selection is sensitive to algorithm parameters and that, by selecting algorithms that produce biologically relevant results, we can improve the efficiency of detecting new biomarkers. The omniBiomarker system, a web-based application which aims to implement the knowledge-guided workflow, allows users to simultaneously test several algorithms for feature ranking and selection from high-throughput biological data (right panel).

parameters, and 5) efficient analysis with a parallelized computing back-end. We designed these functions as modules within a multi-tiered application (**Figure 54**). The top-level tier of the application includes the web browser, where all user interactions take place. Here, users may upload and download gene expression data as well as gene ranking results. The web interface also provides intuitive controls for manipulating and organizing data and results. All users must login prior to any data access or manipulation. The second tier of the application includes the logic for displaying the user interface. In addition, this layer contains simple utility applications for data normalization and access/retrieval from the database, which is the third tier. The second web-server tier may also directly access the computation layer, which contains modules for gene ranking. The computation layer accesses the database layer in order to retrieve gene expression data and to store analysis results.

omniBiomarker is available for public use. However, access to the system requires that users register to create a user profile. Once a user has created a profile, he or she may proceed to upload gene expression data and begin data analysis (**Figure 55**). The system accepts gene expression data in either Microsoft Excel or tab-delimited formats such that each column represents a single microarray sample and each row represents a gene. The data files may optionally contain information about sample names or gene names. Gene names are stored in a database under a user-defined microarray name so that data duplication may not occur for future data uploads using the same platform. Once logged in, users are presented with an overview of previously uploaded microarray datasets listed with key pieces of information (**Figure 56**). This information includes the user-specified dataset name, a brief dataset description, the number of samples uploaded, the number of probesets in the microarray, and the date and time of upload. From here, users may delete datasets, or select a dataset to proceed to more a detailed analysis or data management interface

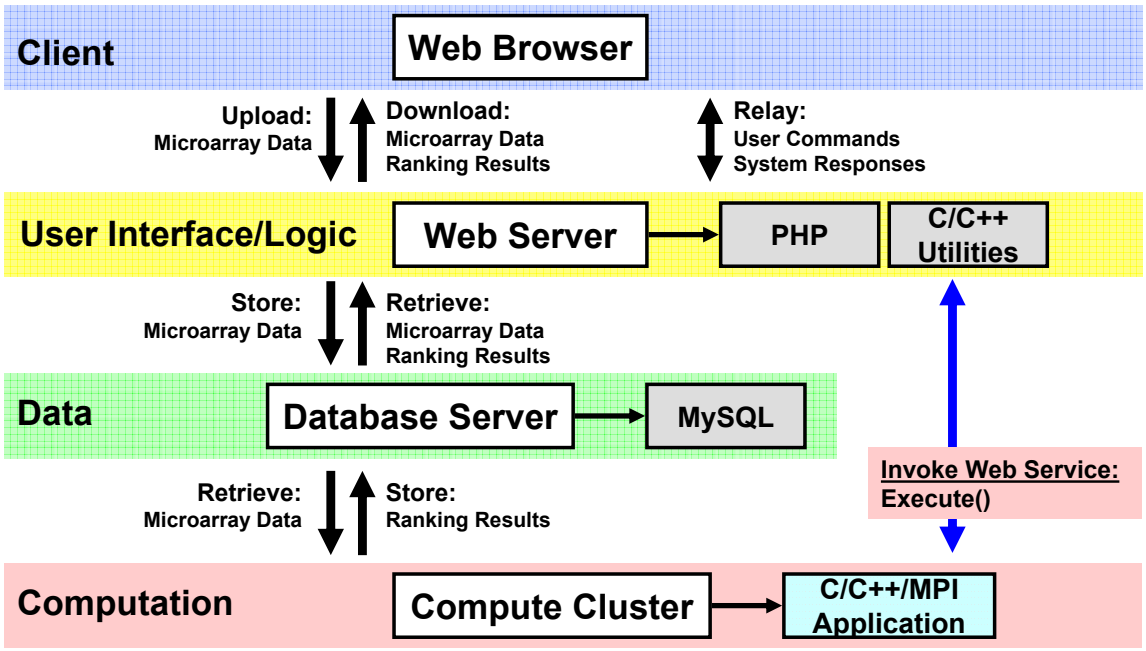


Figure 54: The omniBiomarker web-based application contains four layers. The client layer contains users' web browsers and collects input to be relayed to the web server. The web server layer, in addition to responding to user commands and sending the appropriate interface commands, contains utilities for uploading and downloading data to and from the MySQL relational database. The database is the third layer, accessed by both the web server and computation layers. The computation layer receives commands directly from the web server layer through a web service.

Table 22: Normalization functions available in omniBiomarker. Multiple normalization steps may be applied sequentially in a single procedure.

omniBiomarker Normalization Functions
Column Mean Centering
Column Median Centering
Column Z-score
Row Mean Centering
Row Median Centering
Row Z-score
Log Base e Normalization
Log Base 10 Normalization
Log Base 2 Normalization
Replace Zeros
Replace Zeros and Negatives
Quantile Normalization

(**Figure 57**). omniBiomarker gives the user flexibility in terms of sample partitioning for biomarker identification. After uploading a gene expression dataset, users are presented with the default sample partitioning according to the labels in the original data file. However, users may also re-partition datasets by re-labeling samples. For example, suppose a user uploads a cancer gene expression dataset that contains three disease subtypes. The user might be interested in identifying biomarkers that are differentially expressed between subtypes A and B. In this case, the user would need to create a new data partition, move samples from classes A and B into the new partition, and re-label the samples (using -1 and 1). The new partition refers to the original samples and does not duplicate data samples, reducing the overall storage overhead of the database. omniBiomarker also includes a function to automatically create stratified n-fold cross validation partitions of datasets in order to compute an estimate of predictive performance.

From the data management interface, users may also apply normalization procedures to the data (**Table 22**). The normalization procedures include common

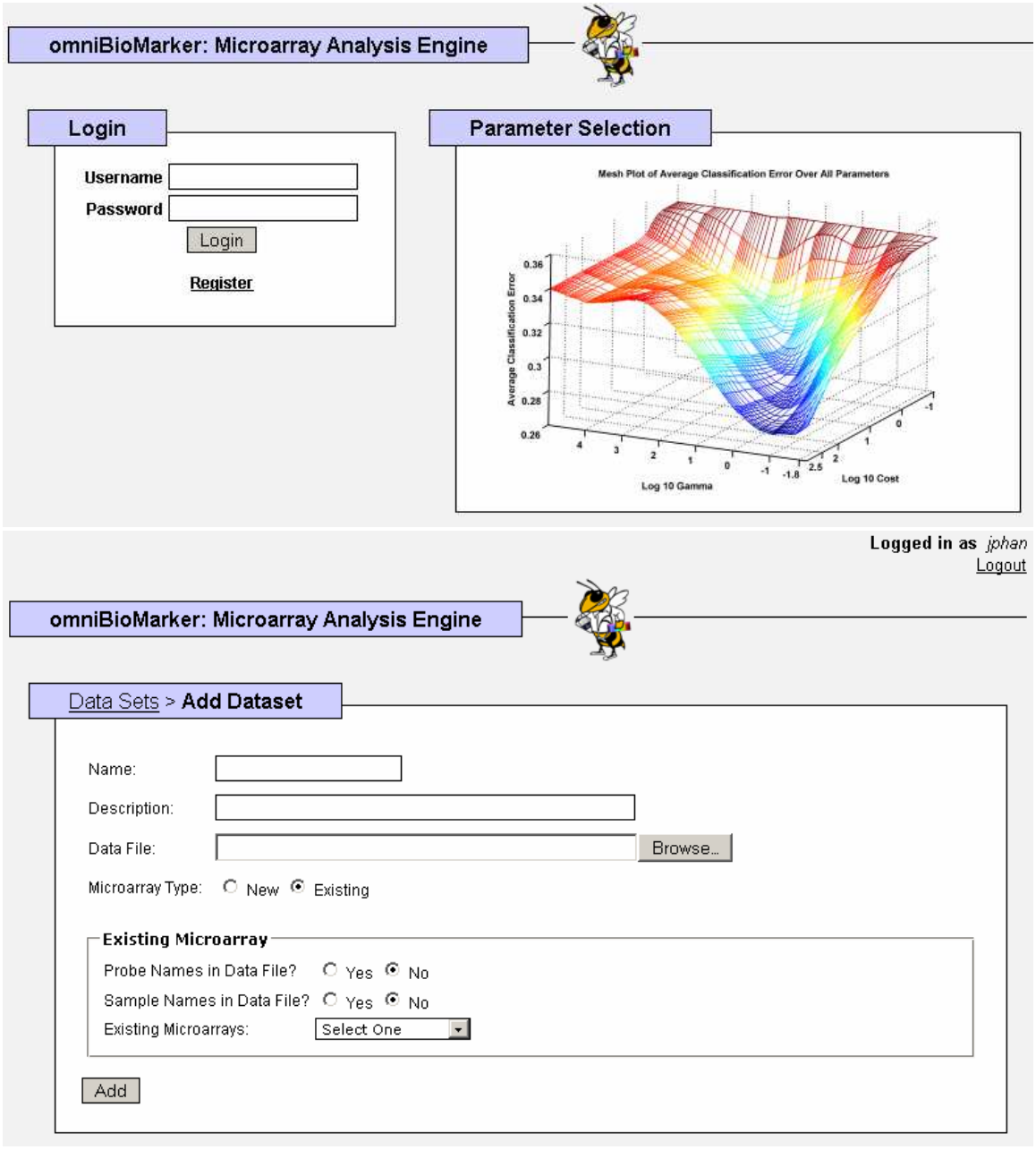


Figure 55: omniBiomarker interfaces for login access and data upload. Users must register before accessing the system through the login screen (top). After logging in, users are presented with a list of datasets that they have previously uploaded. New users will not see any datasets. Users can upload data using a web-form (bottom).

Name		Partitions	Samples	Probesets	Microarray	Creation Time	Users	Delete
Prostate_ChandranNorm_TvsAdj	☐	101	124	12625	Affy_HG_U95Av2	2009-02-19 11:17:03		
	■	Prostate_ChandranNorm_TvsAdj			Custom			
Prostate_SinghNorm_TvsAdj	☐	101	102	12625	Affy_HG_U95Av2	2009-02-10 20:36:53		
	■	Prostate_SinghNorm_TvsAdj			Custom			
Breast_MDACC2_pCRvsRD	☐	101	49	22283	Affy_HG_U133A	2009-02-09 12:32:09		
	■	Breast_MDACC2_pCRvsRD			Custom			
Breast_MDACC1_pCRvsRD	☐	101	81	22283	Affy_HG_U133A	2009-02-09 12:31:30		
	■	Breast_MDACC1_pCRvsRD			Custom			
Prostate_Chandran_TvsAdj	☐	101	124	12625	Affy_HG_U95Av2	2009-02-01 14:10:04		
	■	Prostate_Chandran_TvsAdj			Custom			
Prostate_Singh_TvsAdj	☐	101	102	12625	Affy_HG_U95Av2	2009-02-01 03:19:09		
	■	Prostate_Singh_TvsAdj			Custom			
Renal_Jones_CCvsPAP	☐	101	43	8793	Affy_HG_Focus	2009-01-31 14:16:35		
	■	Renal_Jones_CCvsPAP			Custom			
Renal_Schuetz_CCvsPAP	☐	101	18	8793	Affy_HG_Focus	2009-01-30 20:59:59		
	■	Renal_Schuetz_CCvsPAP			Custom			
Renal_Jones_CCvsONCCHR	☐	101	50	8793	Affy_HG_Focus	2009-01-30 15:37:34		
	■	Renal_Jones_CCvsONCCHR			Custom			
Renal_Schuetz_CCvsONCCHR	☐	101	20	8793	Affy_HG_Focus	2009-01-30 15:36:00		
	■	Renal_Schuetz_CCvsONCCHR			Custom			

Figure 56: omniBiomarker data list interface. The dataset list displays standard information about the particular dataset, including data name, a brief description, number of samples, number of probesets in each microarray sample, the microarray type, and the upload date. From this screen users may select a particular dataset for further analysis or delete the dataset by clicking the red 'X' icon.

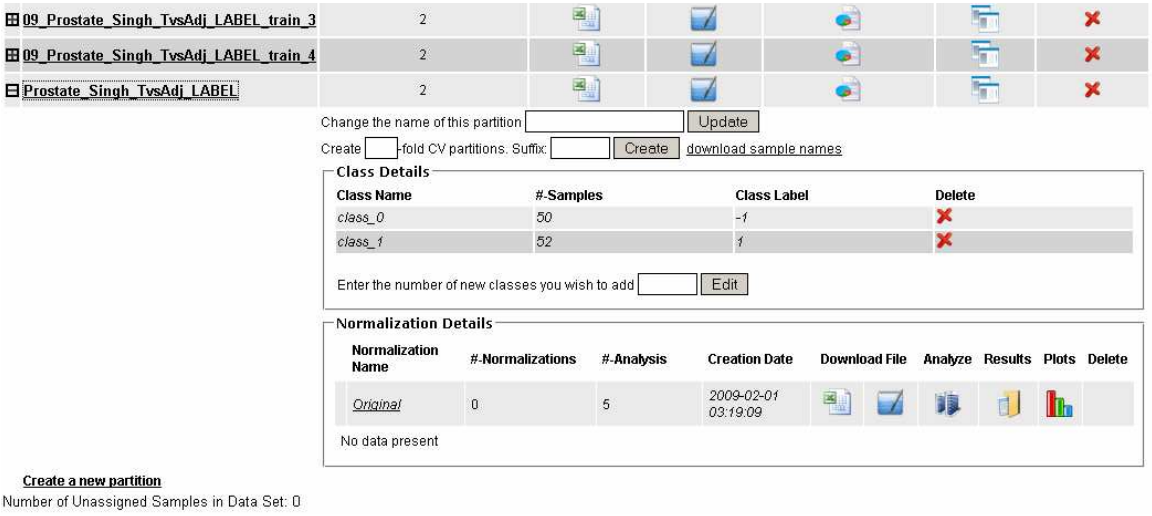


Figure 57: omniBiomarker interface for data and sample management. From this data management interface, users may re-partition microarray samples into different groups prior to feature selection analysis. Additionally, users may apply preprocessing normalization steps to the data, download the normalized data, and initiate feature selection analysis on the normalized data. This interface also includes a function to automatically partition the data for stratified n-fold cross validation.

transformations such as column and row mean centering, median centering, or z-score (mean centering and division by the standard deviation). omniBiomarker also includes simple procedures such as log transforms and replacement of missing values as well as more sophisticated methods such as quantile normalization. Quantile normalization forces all samples in the dataset into a common mean distribution [7]. Multiple normalization procedures may be applied sequentially to each dataset in a single step.

From the data management interface, users may select a dataset–or a normalized version of a dataset–and proceed to the feature ranking analysis interface. omniBiomarker provides primarily wrapper-based feature ranking methods. Wrapper-based methods rank features by computing an estimate of classification performance via a cross validation or bootstrapping procedure [10, 151]. This procedure is generally more computationally intensive compared to filter methods such as fold change or SAM [141]. However, wrapper-based methods are desirable when the end goal is to

construct a clinical predictor, as the features are selected based on estimated prediction performance. Wrapper-based methods are highly parametric. They vary in terms of classification method as well as error estimation method. Furthermore, each classifier is subject to its own parameters–including kernels and associated parameters–that need to be tuned. Although cross validation and bootstrap error estimation methods are well known as unbiased estimators, some studies have shown that these methods may not perform as expected in small-sample conditions [10, 44]. Thus, we provide a variety of error estimation methods, including the extremely biased, but very efficient resubstitution method. **Figure 58** is a screenshot of the omniBiomarker feature ranking analysis interface. In addition to classifier and error estimator options, omniBiomarker provides parameters to tune computational efficiency. omniBiomarker allows us to rank a gene expression dataset using multiple parameters simultaneously. For example, submitting N parameter combinations into the ranking system will rank the dataset N times. After processing, users may download and examine all N rankings to identify the most biologically relevant. Multi-parameter ranking analysis is very computationally intense. As such, omniBiomarker partitions the ranking procedures across multiple compute nodes based on the data distribution parameters. All analysis parameters for omniBiomarker are listed in **Table 23**. After selecting analysis parameters and submitting the job, the user may control the job from the analysis results and queuing interface (**Figure 59**). As the ranking proceeds, the user can observe the job status and download results once analysis has completed. The downloaded results are formatted such that users can correlate ranking analysis parameters with ranking results, facilitating parameter tuning based on the quality of results.

All omniBiomarker data are stored in a relational database that organizes information about expression values as well as analysis results (**Figure 60**). Microarray samples are stored in a multi-level hierarchy in order to the maximize flexibility of

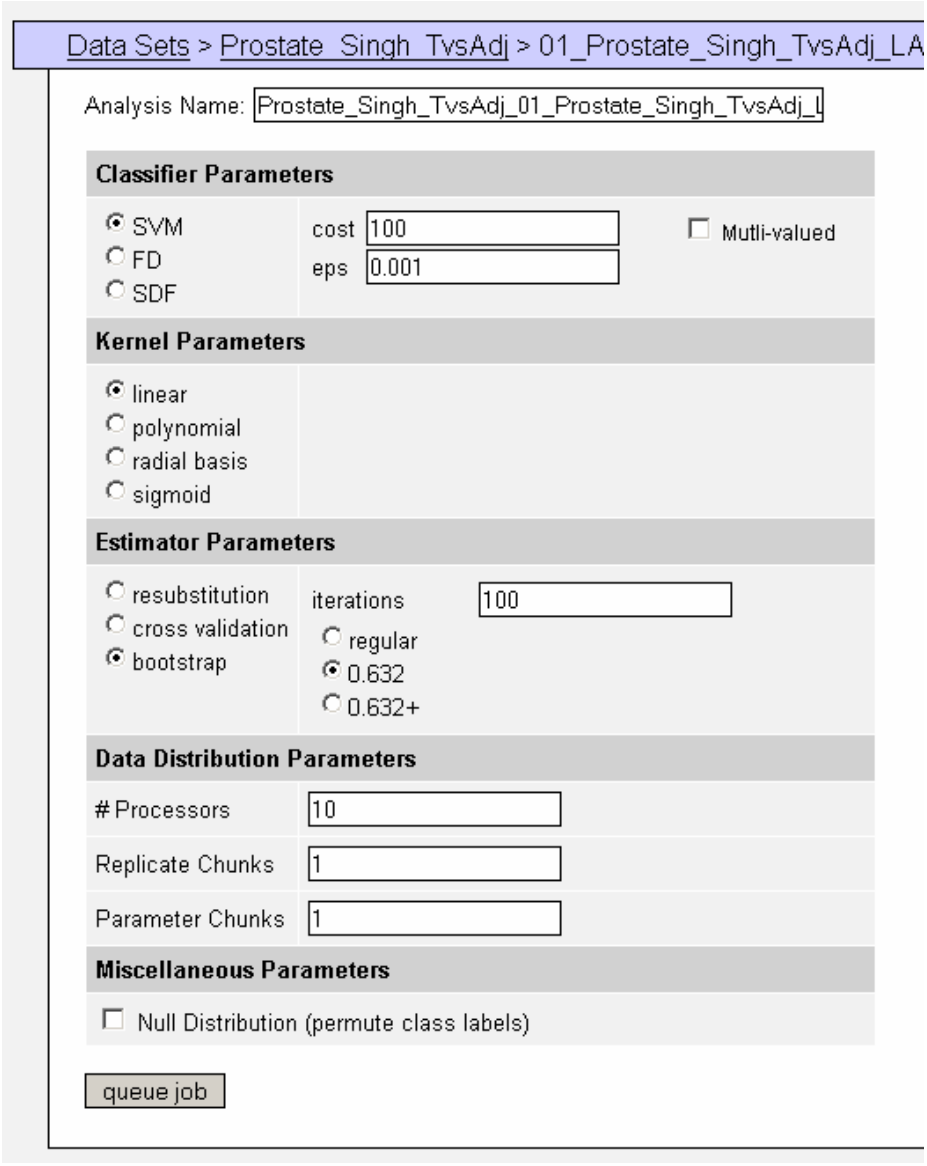


Figure 58: omniBiomarker gene ranking analysis form. This interface provides the user with several wrapper-based ranking methods that include three classifiers, four classifier kernels, and several error estimation methods. In addition, users may also tune the distributed computing parameters.

Table 23: Analysis options in the omniBiomarker web-based application.

omniBiomarker Analysis Options	
Classifier	**Description/Parameters**
Support Vector Machine (SVM)	SVM cost, EPS (machine precision)
Linear/Fishers Discriminant (FD)	Regularization Factor (gamma)
Signed Distance Function (SDF)	Regularization Factor (gamma)
Classifier Kernel	**Description/Parameters**
Linear	N/A
Polynomial	Coefficient, Degree, Gamma
Radial Basis	Gamma
Sigmoid	Coefficient, Gamma
Error Estimation Method	**Description/Parameters**
Resubstitution	N/A
Cross Validation	# of Iterations, # of Folds
Bootstrap	# of Iterations, Regular/0.632/0.632+
Additional	**Description/Parameters**
# of Processors	# of processors to distribute job
Replicate Chunks	# of genes/rows distributed to each CPU
Parameter Chunks	# of parameters distributed to each CPU

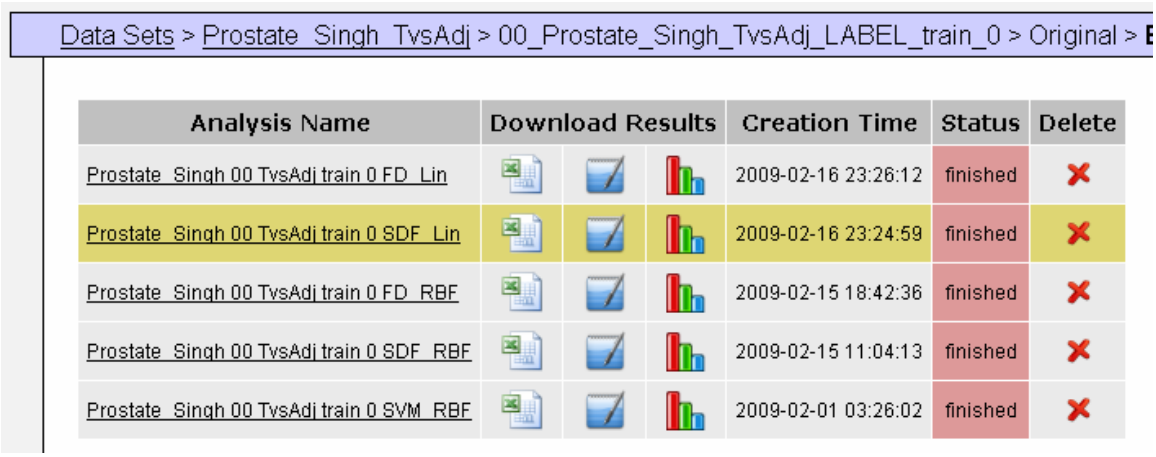


Figure 59: omniBiomarker interface for analysis results and job queue. This interface reports the status of all submitted feature ranking jobs. Once analysis has completed for a job, users may download the results in either Microsoft Excel or tab-delimited text format.

data analysis while reducing overall storage requirements. A dataset typically consists of several microarray samples partitioned into specific phenotypic classes. These partitions are called 'label sets' and serve to reduce data duplication. Each dataset is associated with metadata tables so that each biomarker can be directly linked to the appropriate annotation information. In addition to sample storage tables, the database also includes several tables to store biomarker rankings, analysis parameters, and knowledge about biomarker validation results.

Information about the dataset platform (type of microarray chip, etc.) is stored in a separate metadata table. Datasets can be normalized or pre-processed in a variety of ways. The normalized data is stored in a separate table. There is always at least one normalized dataset for each dataset. The default normalized data is called the 'original' data. If we want to normalize the original data by, for example, scaling all expression values, we would create a new normalization record. As a result, we would then have two normalized versions for the dataset: the original and the normalized.

Each normalized dataset is linked to multiple items in the 'orig_sample' table. Each record in the 'orig_sample' table is a unique microarray sample, containing up to thousands of gene expression values. We want to be able to partition the samples into different classes without duplicating data. The following is an example of a data re-partitioning scenario:

Suppose that a microarray dataset has several samples: (1 2 3 4 5 6 7 8 9). If samples (1 2 3 4) are microarray samples from patients that exhibit a specific symptom and samples (5 6 7 8 9) are from patients that have no symptoms, then we want to compare the two groups (1 2 3 4) and (5 6 7 8 9) to identify biomarkers that are differentially expressed between these groups. Then, according to the relational database diagram, the 'data_set' entry will be linked to one 'label_set' entry that will be named 'partition1', for example. The 'partition1' 'label_set' will be linked to two items in the class table: 'class1' for (1 2 3 4) and 'class2' for (5 6 7 8 9). Each

of these class records will be linked to 4 and 5 unique records in the sample table, respectively. Finally, each record in the sample table will be linked to a single record in the 'orig_sample' table. The 'orig_sample' record is what contains the actual data for the microarray sample.

Now suppose that physicians discovered another symptom to distinguish these patients and the symptom is present in patients (1 2 3 4 5 6) but not in patients (7 8 9). We are also interested in identifying differentially expressed genes between these two groups. In this scenario, we define another record in the 'label_set' table called 'partition2' that is linked to two more items in the class table called 'class1' and 'class2'. Although these class names are the same as the previous partitioning, the 'class_id' and 'label_set_id' values will be different. These class records will be associated with another unique set of records in the sample table, but these samples point back to the same items in the 'orig_sample' table as the 'partition1' 'label_set'. Using this data scheme, omniBiomarker can efficiently store gene expression data with minimal data duplication.

6.3 omniBiomarker: caBIG Grid Services

Web-based omniBiomarker is usable as a standalone application. Users may upload data in a simple format and retrieve results in a similar format. However, as with many other bioinformatics applications, these data formats may not be compatible with other similar applications. Because omniBiomarker primarily provides feature selection functionality, it would be natural to expect users to use omniBiomarker in conjunction with other applications. For example, prior to feature selection, users may need to assess the quality of their microarray data with an application such as caCorrect [137]. The data format that caCorrect produces is similar, but not compatible with the data format required by omniBiomarker. Similarly, omniBiomarker produces, as output, a list of genes in the original data order with a corresponding

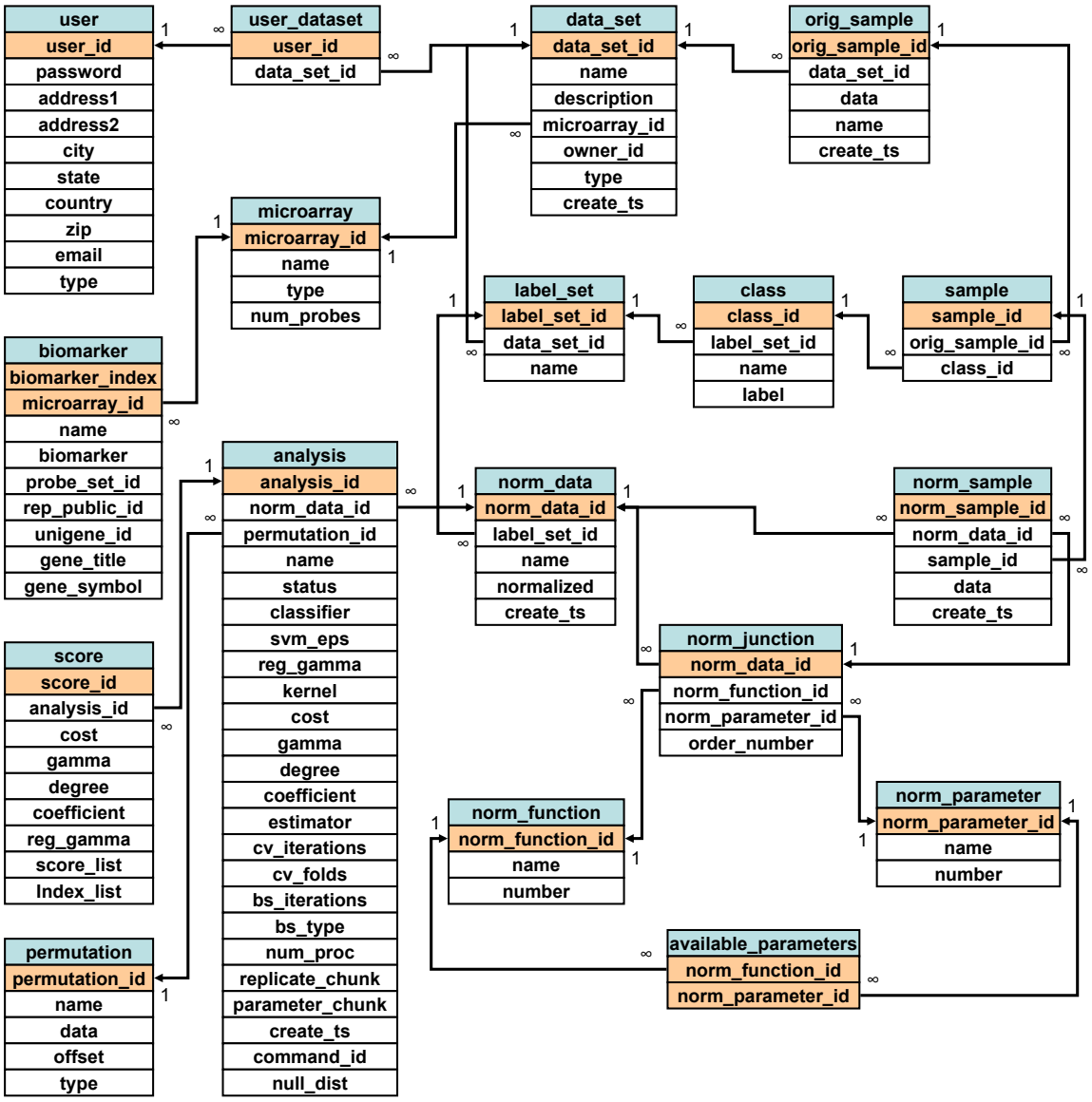


Figure 60: omniBiomarker web-based application relational database. This database is designed to store microarray data as well as feature selection results. Microarray data are stored in a hierarchy that allows users to normalize data and assign samples into classes for supervised analysis. The 'analysis' table stores all parameters for a particular feature ranking job as well as the ranking results (linked with the 'score' table) so that users may assess the results from multiple ranking analyses and select the most biologically relevant result.

list of rank scores. Users can manually filter this data to extract the top genes for further interpretation with an application such as GOMiner [154]. Of course, the developers of omniBiomarker can simply change the data input and output requirements. However, there are many existing bioinformatics applications and developers may not feel the need to alter their applications. Furthermore, users may not want to use caCorrect or GOMiner. They may want to use dChip or GOStat for their microarray quality control and gene function interpretation needs, which may very well require completely different data formats [78, 4]. As such, we see a significant lack of interoperability among bioinformatics applications.

Recently, the National Cancer Institute (NCI) has initiated the Cancer Bioinformatics Grid (caBIG) project, which attempts to tackle the problem of interoperability between bioinformatics applications. Specifically, they focus on bioinformatics research in the area of oncology and developed a semantically interoperable infrastructure for bioinformatics applications [95, 87]. The primary function of caBIG is to define a set of standards and control a centralized vocabulary for potential caBIG-compatible applications. This standardization forces software applications to reuse data structures and vocabulary, increasing overall interoperability [74, 99]. Each application that intends to become caBIG-certified must be subject to a rigorous review process in which every data element that is part of the application's input or output routines is mapped to a predefined semantic vocabulary term [26, 42]. caBIG is a significant undertaking because of the existence of many bioinformatics applications–many of which will not become certified–as well as the difficult task of defining controlled vocabulary terms for each bioinformatic data element. Indeed, the lack of software interoperability is not constrained to only the bioinformatics research community. The ramifications of a fully interoperable infrastructure may also improve health care informatics in general [73].

We have begun the process of certifying components of omniBiomarker as caBIG

compatible. caBIG requires that the primary and publicly available data analysis routines of a bioinformatics application be exposed as a web service. Web services are well-defined application programming interfaces (API) that can be accessed by any application that knows the required input parameters and expected output. The omniBiomarker web-based application contains many functions for data storage, pre-processing, and feature selection. However, we only translate basic feature ranking functions into the caBIG standard due to the availability of other caBIG-certified applications that serve as data repositories such as caArray. Therefore, the caBIG-certified omniBiomarker application no longer stores microarray data. Rather, it retrieves all data for analysis from caArray. We implement two basic functions for feature ranking: rankGenes() and getGeneRanks() (**Table 24**). The input parameter for the rankGenes() function specifies the dataset, classifier, and error estimation method for feature ranking. This function returns a reference object, which may be passed to the getGeneRanks() function to retrieve the list of ranking results once analysis is complete.

The overall structure of the caBIG omniBiomarker system is relatively simple compared to the original web-based application (**Figure 61**). Since only one set of ranking parameters may be executed at once, the compute cluster is no longer required, reducing the number of system layers to three. The web services can be invoked by any software client that understands the web service interfaces. Typically, the client application is a Java-based and also provides an interface for user interaction. The application layer includes caGrid system functions as well as the C/C++ application that performs the actual gene ranking. Gene ranking analysis parameters and results are stored in a simplified relational database that only contains a single table. Results are stored indefinitely and may be retrieved as long as the analysis ID is known.

Table 24: caBIG service interfaces for omniBiomarker.

omniBiomarker caBIG Service Interfaces		
Name	**Input**	**Output**
rankGenes	GeneRankParameters	GeneRankAnalysis
getGeneRanks	GeneRankAnalysis	GeneRankResults

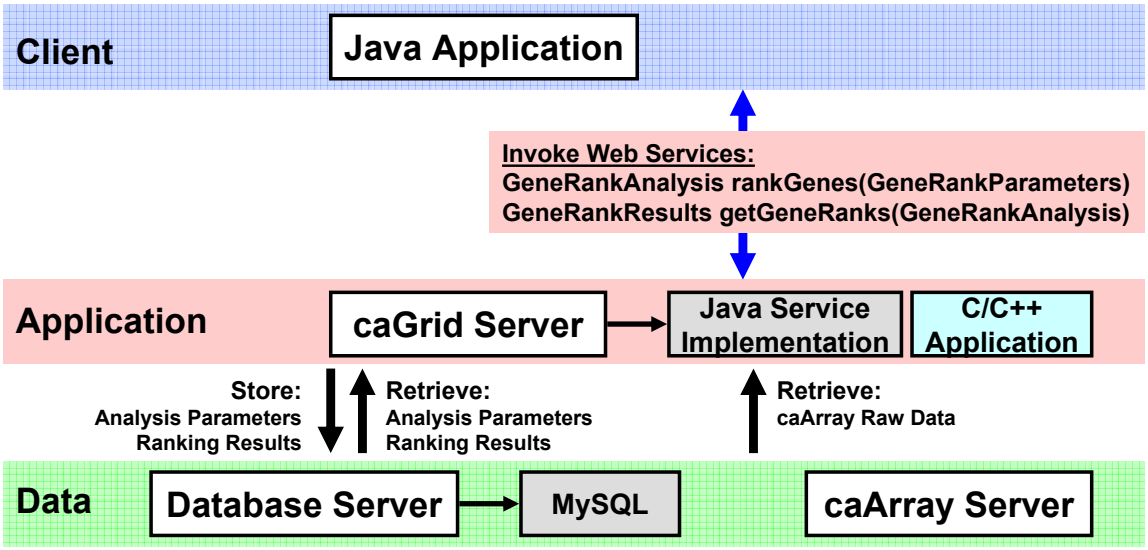


Figure 61: omniBiomarker caBIG System. In addition to semantic standardization, conversion of omniBiomarker to a caBIG-enabled set of grid services simplifies the overall system architecture. Client applications access the services through two caBIG web services, rankGenes() and getGeneRanks(). These web services initiate a gene ranking analysis and retrieve the ranking results, respectively. Ranking results are still stored in a simplified MySQL relational database. However, all microarray data are stored in the standardized caArray server.

The caBIG review process for certification requires that developers create well-defined object models (UML diagrams) for each data class that is passed to and from their service functions. The omniBiomarker rankGenes() function requires the GeneRankParameters class as input (**Figure 62**). This class contains several other classes that specify the classification and error estimation method used for wrapper-based feature ranking. Additionally, the getGeneRanks() function returns the GeneRankResults class, which contains information about analysis parameters as well as the ranking results (**Figure 63**).

caBIG also requires that each data element within the UML diagram have well-defined vocabularies [26]. These vocabulary terms are centrally stored and managed by caBIG such that future applications may share the same terms, resulting in an overall increase in application interoperability. We defined terms for omniBiomarker that pertain to wrapper-based feature ranking methods (**Table 25**). Because omniBiomarker is one of the first analytical applications in caBIG, these terms were defined specifically for omniBiomarker, but may now be used by other applications.

6.4 Conclusion

omniBiomarker is a web-based bioinformatics application that serves as both a microarray data repository as well as a tool for identifying clinically useful candidate biomarkers. The theoretical foundation of omniBiomarker addresses the ill-posed problem of biomarker identification by tuning steps in the analysis pipeline using prior biological knowledge. It is essential that all the analytical steps in the biomarker identification process be recorded in order to later identify the parameters that are most likely to result in biologically relevant solutions. The omniBiomarker system provides users with a data storage and interpretation interface that enables easy tracking and visualization of feature ranking results. Furthermore, the application recognizes the need for high-performance computing in order to allow users to assess

a large population of feature ranking algorithms simultaneously and efficiently.

omniBiomarker also addresses the problem of community accessibility. It is developed with a focus on not only the novelty of the analysis pipeline, but also on the integration of these analytical steps into a user-friendly, web-accessible interface. This attribute of bioinformatics tools has become increasingly important as the gap between clinical applications and bioinformatics narrows. omniBiomarker is also caBIG-compatible, further increasing the interoperability of its functions with other bioinformatics tools in the cancer research community.

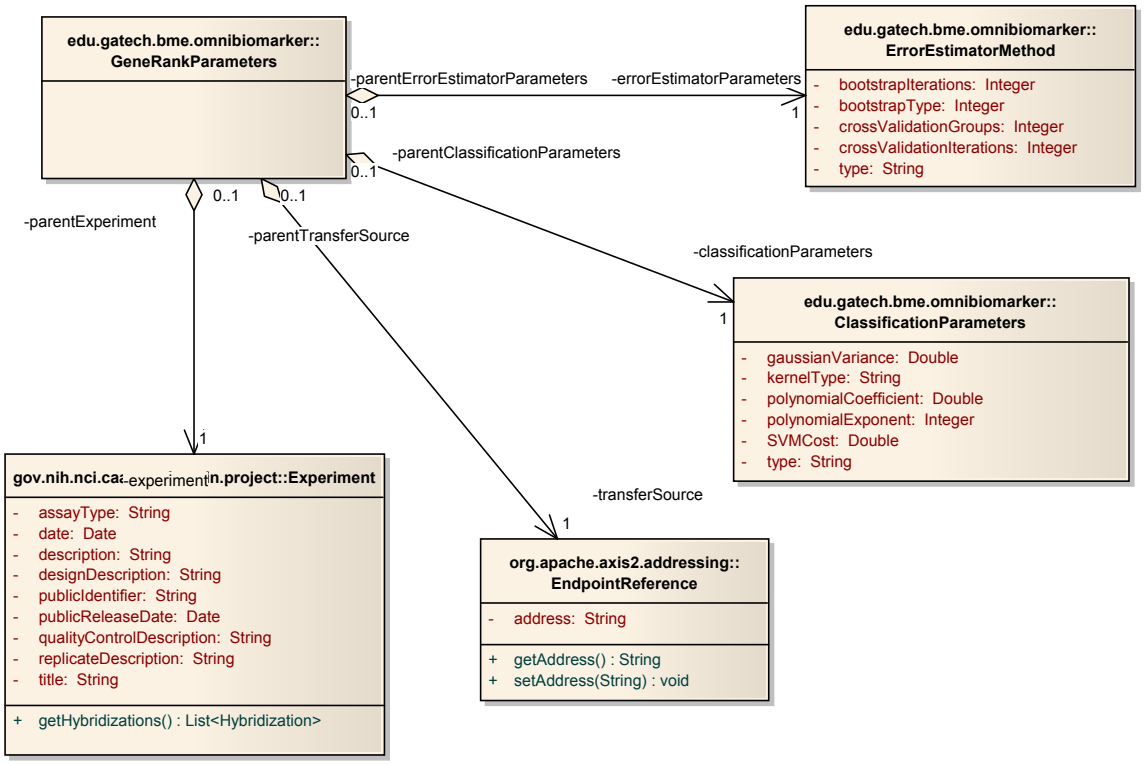


Figure 62: omniBiomarker UML diagram for the GeneRankParameters class. This class is a required parameter for the rankGenes() caBIG web service.

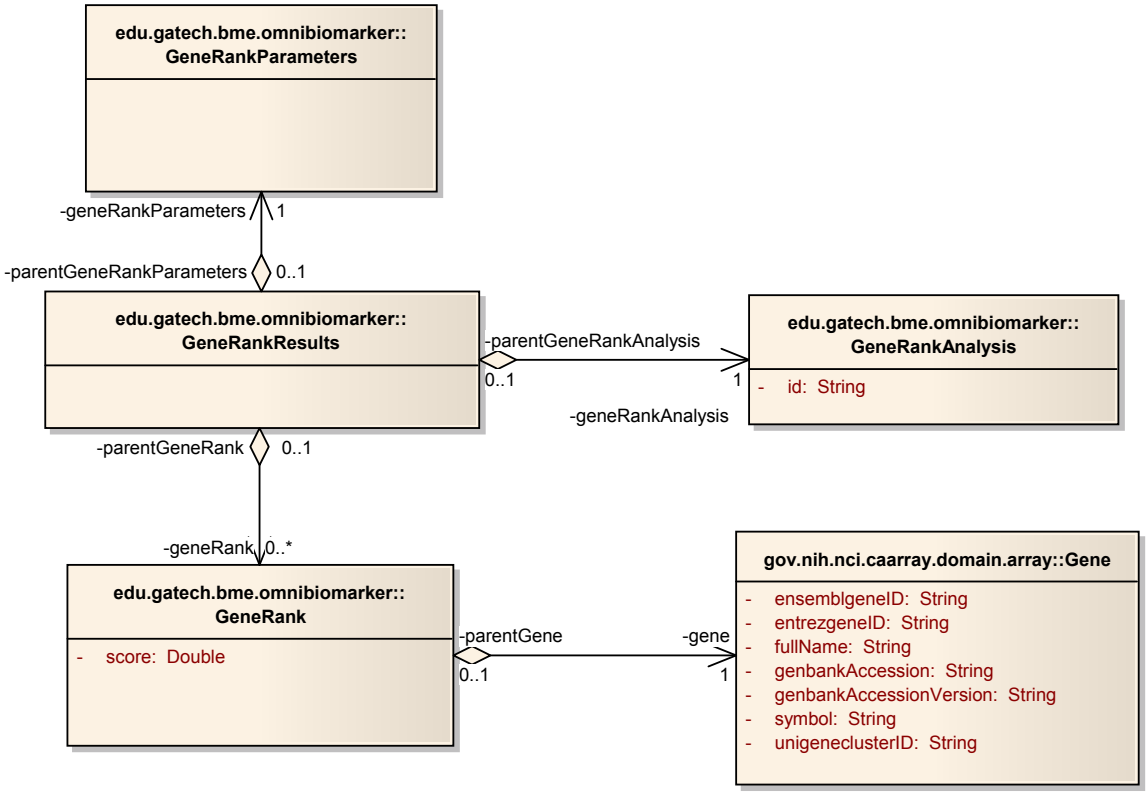


Figure 63: omniBiomarker UML diagram for the GeneRankResults class. This class is returned from the getGeneRanks() function and includes information about the analysis parameters as well as the analysis results.

Table 25: caBIG controlled vocabulary terms for omniBiomarker.

Term (Synonym)	**NCI Thesaurus Code**	**Definition**
Gaussian Distribution (Normal Distribution)	C78534	A family of continuous probability distributions defined by the parameters of mean and variance.
Polynomial (Polynomial Expression)	C78538	A mathematical expression formed by summing multiples of powers of some variable.
Support Vector Machine (SVM)	C78542	A classification method that will construct a hyperplane between two sets of data such that it maximizes the margin between the hyperplane and the nearest samples.
Kernel Function	C78544	A function that measures the distance between two expression vectors as the data are projected into higher-dimensional space.
Cross Validation	C78545	A statistical method of partitioning a sample of data into subsets such that the analysis is initially performed on a single subset, while the other subsets are retained for subsequent use in confirming and validating the initial analysis.
Bootstrapping (Bootstrap)	C78547	A method for estimating properties of a dataset by measuring those properties when sampling from an approximating distribution.
Cost Function	C78559	A function or expression that is either maximized or minimized in order to obtain the optimal solution for a mathematical problem.
Optimization	C78561	A process that attempts to find the best solution to a problem.

CHAPTER VII

CONCLUSION

Biomarkers are essential for the successful treatment of cancer since they enable early detection of the disease before significant symptoms arise [156]. Moreover, pathologists may use biomarkers to acquire information from tissue biopsies that may not be readily apparent using traditional examination techniques. A cancer detection screening using biomarkers is essentially a clinical predictor that assigns patients to categories of disease presence/absence or degree of disease severity. Accurate assignment of patients into these categories would optimize therapeutic decisions and improve treatment success rates.

Before we can use biomarkers in the clinic, however, we must subject them to a rigorous validation process. This process of biomarker discovery and clinical application is commonly referred to as translational research [112]. Indeed, there are many existing candidate biomarkers due to the advent of high-throughput assay platforms such as gene expression microarrays. Correspondingly, we have seen a sharp increase in bioinformatics applications and algorithms designed to process the resulting data from high-throughput technology. However, without proper validation, many of these candidate biomarkers may never be used in a clinical setting. Thus, the number of candidate biomarkers is vastly greater than the number of biomarkers actually applied to patients. Validation of a biomarker is a confirmation of accuracy, reproducibility, and effectiveness in detecting disease [156]. Gene expression microarrays, the predominant method for detecting candidate biomarkers, suffer from low specificity of detections due to the high-dimensionality of the data. Validation technologies such

as qRT-PCR, protein microarrays, and tissue microarrays are generally sensitive, reproducible, and accurate. After laboratory validation, however, these biomarkers are still subject to rigorous clinical testing prior to final application.

When applying the multitude of existing bioinformatics algorithms to a highly variable population of clinical data, we must consider two well-known machine learning theories: the 'No Free Lunch' and the 'Ugly Duckling' theorems. The 'No Free Lunch' theorem states that there is no single machine learning algorithm that performs well for all datasets [32]. In the domain of clinical prediction, this theorem implies that a particular set of classification parameters and feature selection methods may be appropriate for a specific clinical dataset, but not for others. The related 'Ugly Duckling' theorem applied to bioinformatics states that domain-specific knowledge plays an important role in identifying informative feature sets. Thus, there is no problem-independent feature selection method that can be generally applied to all problem domains successfully [32]. The heterogeneity of clinical data and the availability of many bioinformatics algorithms for feature selection and classification pose a difficult problem. Feature selection and clinical prediction will generally lack reproducibility when applying similar procedures to different datasets, even if the datasets are clinically similar.

Fortunately, the availability of validated biomarkers adds an extra dimension to bioinformatics algorithms. Rather than only relying on high-throughput data that may suffer from technical as well as biological variability, we may use these validated biomarkers as domain knowledge to guide steps in the analytical pipeline. During the feature selection process, biological knowledge can narrow the space of relevant feature selection algorithms and improve the efficiency of detecting novel biomarkers that are likely to validate [106]. Furthermore, feature selection is an essential step in building clinical predictors. The results presented in **Chapter 5** indicate that the reproducibility and accuracy of clinical predictors improves when we narrow the space

of relevant feature selection methods using prior knowledge.

Finally, the reproducibility of bioinformatics results suffers from a lack of interoperability between various bioinformatics software applications. We have designed an application, called omniBiomarker, to facilitate the guidance of biomarker identification and, eventually, clinical predictor assessment. This application improves reproducibility by tracking all analytical steps so that users may visualize and identify optimal problem-specific parameters. Moreover, omniBiomarker is also designed to be compatible with the NCI Cancer Bioinformatics Grid (caBIG) in order to ensure that the application is semantically interoperable with other bioinformatics applications. caBIG compatibility also ensures that omniBiomarker is accessible to a wide community of scientists that are interested in knowledge-guided biomarker identification and clinical prediction.

APPENDIX A

MODELING KNOWLEDGE IN BIOMARKER IDENTIFICATION

In **Chapter 4**, we defined gene expression datasets using the variables $\vec{x}_n^i$ and y_n that represent an observation n for feature set i and a class label. These variables are single observations from the random variables $\vec{X}^i \in \mathbb{R}^p$ and $Y \in \{0, 1\}$. Recall that p represents the number of genes in each gene set and is much smaller than ℓ, which is the total number of genes in a dataset ($p < \ell$). $\vec{X}^i$ and Y are jointly distributed. We also defined $\vec{d}_i = ((y_1, \vec{x}_1^i), (y_2, \vec{x}_2^i), \ldots, (y_N, \vec{x}_N^i))$ as a vector of all samples from a gene set i. $\vec{d}_i$ is a single observation from the random variable $\vec{D}$.

The function $\alpha_i = h_\theta(\vec{d}_i)$ assigns to each gene set a score that represents the differential expression of that gene. α_i is a single observation of the random variable A. The variable $\theta \in \Theta = \{\theta_1, \theta_2, \ldots, \theta_\vartheta\}$ is a meta-parameter that characterizes the ranking function. Gene set ranking results are highly sensitive to θ. Thus, we want to identify the θ that produces the most biologically relevant ranking according to prior knowledge specific to the dataset. We can define prior knowledge for each gene set using the variable r_i, an observation of the random variable $R \in \{0, 1\}$. r_i represents the biological relevance of the gene set i to the ranking problem. Generally, we want gene sets with $r_i = 1$ to be ranked more favorably than gene sets with $r_i = 0$. The random variables $\vec{D}$ and R are jointly distributed.

Since $\vec{D}$ and R are jointly distributed and the distribution of A is dependent on both $\vec{D}$ and the ranking method, θ, then A, R, and θ are also jointly distributed.

Thus, we can define conditional density functions for A as

$$f_{0,\theta}(\alpha) = f(\alpha|\theta, R = 0) \tag{21}$$
$$f_{1,\theta}(\alpha) = f(\alpha|\theta, R = 1)$$

$f_{0,\theta}(\alpha)$ represents the distribution of rank scores for all biologically irrelevant gene sets and some ranking metric, θ. Likewise, $f_{1,\theta}(\alpha)$ represents the distribution of rank scores for all biologically relevant gene sets. In the following sections, we derive methods for examining these distributions in order to choose the most biologically relevant ranking metric, θ. We can use maximum likelihood or maximum *a posteriori* methods to choose θ.

The choice of ranking method, θ, affects the distributions of $f_{0,\theta}(\alpha)$ and $f_{1,\theta}(\alpha)$. Assuming that lower α indicates a more differentially expressed gene set, the most biologically relevant θ should correspond to distributions of $f_{0,\theta}(\alpha)$ and $f_{1,\theta}(\alpha)$ that are maximally separated with the expectation of $f_{1,\theta}(\alpha)$ less than that of $f_{0,\theta}(\alpha)$. The biological relevance of θ is equal to the probability that observations from the $f_{1,\theta}(\alpha)$ distribution are less than observations from the $f_{0,\theta}(\alpha)$ distribution. We can compute this probability with

$$\phi(\theta) = \int_0^1 f_{1,\theta}(\alpha) \left[\int_\alpha^1 f_{0,\theta}(x)dx \right] d\alpha, \tag{22}$$

which is equivalent to the area under an ROC curve (AUC). For example, if we define a discriminating threshold τ in the interval [0,1], then the functions

$$x(\tau, \theta) = \int_0^\tau f_{0,\theta}(\alpha)d\alpha \tag{23}$$
$$y(\tau, \theta) = \int_0^\tau f_{1,\theta}(\alpha)d\alpha$$

represent the false positive rate (FPR) and true positive rate (TPR) of identifying relevant gene sets. All gene sets with $\alpha < \tau$ are classified as relevant. $x(\tau, \theta)$ and $y(\tau, \theta)$ trace the receiver operator characteristic (ROC) curve. We can compute the

AUC with

$$AUC(\theta) = \int_0^1 y(\tau, \theta) x'(\tau, \theta) d\tau, \tag{24}$$

which is equivalent to **Equation 22**.

In practice, we do not know the distribution of $f_{1,\theta}(\alpha)$ since our knowledge of the biologically relevant genes in a dataset is limited. In fact, our knowledge set consists of the set of k gene sets, $G_k = \{g_1, g_2, \ldots, g_k\}$, where $k \ll m$ and m is the total number of gene sets. Using this knowledge, we can simplify the biological relevance function, **Equation 22**, to

$$\phi(G_k, \theta) = \frac{1}{k} \sum_{i=1}^{k} \int_{h_\theta(\vec{d}_{g_k})}^{1} f_{0,\theta}(x) dx. \tag{25}$$

We can further estimate the $f_{0,\theta}(\alpha)$ distribution directly from the gene sets that are not in the knowledge set using

$$\phi(G_k, \theta) = \frac{1}{k(m-k)} \sum_{i \in G_k} \sum_{j \notin G_k} I\left(h_\theta(\vec{d_i}) < h_\theta(\vec{d_j})\right) \tag{26}$$

where $I(x)$ is the indicator function that evaluates to one when x is true and zero otherwise. This is a direct empirical estimate of the probability that observations from the $f_{1,\theta}(\alpha)$ distribution are less than observations from the $f_{0,\theta}(\alpha)$ distribution:

$$\phi(G_k, \theta) = P\left(h_\theta(\vec{d_i}) < h_\theta(\vec{d_j})\right) \tag{27}$$

where $i \in G_k$ and $j \notin G_k$.

A.1 Maximum Likelihood (ML) Estimation of the Optimal Ranking Metric

We want to choose a θ that maximizes the biological relevance of ranking with respect to a given set of knowledge. Normalizing **Equation 26** results in a probability mass function for all gene combinations, G_k:

$$P(G_k|\theta) = \frac{\phi(G_k, \theta)}{\sum_{j=1}^{m} \sum_{i=1}^{{}_mC_j} \phi(S_j^i, \theta)} \tag{28}$$

where ${}_mC_j$ is the total number of combinations containing j elements out of m total elements and S_j^i is a gene set that contains the i^{th} combination of j elements. We can now define the likelihood function as

$$L(\theta) = P(G_k|\theta) \tag{29}$$

and identify the most biologically relevant θ with

$$\begin{aligned}\hat{\theta} &= \underset{\theta}{\operatorname{argmax}}\, L(\theta) \\ &= \underset{\theta}{\operatorname{argmax}}\, P(G_k|\theta).\end{aligned} \tag{30}$$

The denominator of **Equation 28** is invariant to θ because it is summed over all gene combinations. Therefore the maximum likelihood becomes

$$\hat{\theta}_{ML} = \underset{\theta}{\operatorname{argmax}}\, \phi(G_k, \theta). \tag{31}$$

A.2 Maximum A Posteriori *(MAP) Estimation of the Optimal Ranking Metric*

We can also identify the most biologically relevant ranking metric, θ, using the maximum *a posteriori* method. If we know from previous experiments that some feature selection methods tend to perform better than others for particular datasets, we can define a prior probability, $P(\theta)$. $P(\theta)$ can also be defined from previously validated gene sets, G_k:

$$P(\theta) = \frac{\phi(G_k, \theta)}{\sum_{\theta' \in \Theta} \phi(G_k, \theta')}. \tag{32}$$

If we are given additional information about n biologically relevant genes, G'_n, then we can update our beliefs about the feature ranking metrics by computing a posterior probability:

$$P(\theta|G'_n) = \frac{P(G'_n|\theta)P(\theta)}{\sum_{\theta' \in \Theta} P(G'_n|\theta')P(\theta')} \tag{33}$$

then using the maximum *a posteriori* (MAP) method to estimate $\hat{\theta}$:

$$\begin{aligned} \hat{\theta}_{MAP} &= \underset{\theta}{\operatorname{argmax}}\, P(\theta|G'_n) \\ &= \underset{\theta}{\operatorname{argmax}}\, P(G'_n|\theta)P(\theta) \\ &= \underset{\theta}{\operatorname{argmax}}\, \phi(G'_n, \theta)P(\theta) \end{aligned} \tag{34}$$

APPENDIX B

CLASSIFICATION METHODS FOR GENE RANKING AND CLINICAL PREDICTION

B.1 Support Vector Machines (SVM)

The support vector machine (SVM) classifier computes a hyperplane that separates groups of samples with maximal margin. In other words, the algorithm optimally places the hyperplane such that its distance to the nearest samples is maximized. A detailed description of the SVM can be found in the book by Cristianini *et al.* [28]. In this work, we use an implementation of the SVM by Lin *et al.* [20].

B.2 Signed Distance Function (SDF)

The signed distance function (SDF) classifier is comparable to the SVM classifier in terms of performance [6]. The classifier fits a linear discriminator to signed distances assigned to each sample in the training set. We can estimate signed distances in a number of ways such that the sign of the distance indicates class label. A simple method for estimating distances computes the smallest distance from a sample to another sample of opposite class. We can then assign each sample x_i a signed distance d_i equal to half of the smallest distance times the class label $(+/-1)$. The original authors of the SDF technical report suggested using least squares to solve the linear relation

$$y = w \cdot x + b \tag{35}$$

where each equation to be solved is

$$d_i = w \cdot x_i + b \tag{36}$$

with $i = 1 \ldots n$ training samples, w and x_i are m dimensional vectors, and m is the number of features. After solving for w and b, we can predict the label for a future test sample, x, with

$$l(x) = w \cdot x + b. \tag{37}$$

The SDF classifier can be kernelized by constructing an nxn kernel matrix

$$K = \sum_{i,j=1}^{n} d_i d_j k(x_i, x_j) \tag{38}$$

where $k()$ can be any of a number of kernels (kernel reference). We can solve the kernelized problem by solving the following system of equations for the vector α.

$$(K + n\gamma I)\alpha = d, \tag{39}$$

where γ is a regularization factor to ensure that the matrix is positive definite, I is the identity matrix, α is a vector of weights, and d is a vector of the n signed distances. γ can usually be assigned a very small value, 0.001.

Once we have solved for the vector α, the signed distance of a future test sample can be computed with

$$g(x) = \sum_{i=1}^{n} \alpha_i k(x_i, x), \tag{40}$$

after which we may determine the class assignment by examining the sign of the distance, $g(x)$.

B.3 Linear Discriminant Analysis (LDA)

Linear discriminant analysis (LDA) maximizes the objective

$$J(w) = \frac{w^T S_B w}{w^T S_W w} \tag{41}$$

where S_B is the between class scatter matrix,

$$S_B = \sum_{C} N_C (\mu_C - \bar{x})(\mu_C - \bar{x})^T \tag{42}$$

and S_W is the within class, or pooled covariance matrix,

$$S_W = \sum_C \sum_{i \in C} (x_i - \mu_C)(x_i - \mu_C)^T. \tag{43}$$

μ_C is the mean of samples within class C, N_C is the number of samples in class C, and $\bar{x}$ is the mean of samples in all classes. We can simplify the maximization of the objective to the following eigenvalue problem

$$S_B w = \lambda S_W w \tag{44}$$

from which the linear discriminant hyperplane is the leading eigenvector of the matrix $S_W^{-1} S_B$. This solution gives us the hyperplane orientation but not the threshold. We can compute the classifier threshold by assuming that the optimal hyperplane passes through a point directly between the two class means. Thus, the final discriminant function is

$$l(x) = w \cdot x + b \tag{45}$$

where

$$b = -w \cdot \mu_m \tag{46}$$

$$\mu_m = \frac{1}{2}(\mu_1 + \mu_2) \tag{47}$$

and μ_1 and μ_2 are the means of the samples in each class.

The LDA classifier can also be kernelized to handle non-linear problems. We can map samples, x, to a different feature space using a function $\Phi(x)$ and a kernel function:

$$k(x_i, x_j) = \Phi(x_i) \cdot \Phi(x_j) \tag{48}$$

We do not need to explicitly evaluate Φx as long as we know the kernel function–e.g., linear, polynomial, radial basis (Gaussian), or sigmoid. In the feature space, we can re-write the LDA problem as

$$\begin{aligned} w_\Phi^T S_B^\Phi w_\Phi &= \alpha^T M \alpha \\ w_\Phi^T S_W^\Phi w_\Phi &= \alpha^T N \alpha \end{aligned} \tag{49}$$

where

$$\begin{aligned} M &= (M_1 - M_2)(M_1 - M_2)^T \\ N &= \sum_{m=1,2} K_m(I - 1_{\ell_j})K_m^T. \end{aligned} \tag{50}$$

M_i is an n by 1 matrix composed of the elements $(M_i)_j = \frac{1}{n_i}\sum_{\ell=1}^{n_i} k(x_j, x_\ell^i)$. n is the total number of samples, n_i is the number of samples in class i, $j = 1 \ldots n$, and x_ℓ^i is the ℓ^{th} sample of class i. K_m is an n by n_m matrix composed of the elements $(K_m)_{ij} = k(x_i, x_j^m)$ where $i = 1 \ldots n$ and $j1 \ldots n_j$. The optimization problem using these kernel matrices is similar to the linear case:

$$J(\alpha) = \frac{\alpha^T M \alpha}{\alpha^T N \alpha} \tag{51}$$

The solution to this optimization problem is the vector α which corresponds to the largest eigenvalue of $N^{-1}M$. The hyperplane in the new feature space is

$$w_\Phi = \sum_{i=1}^{n} \alpha_i \Phi(x_i). \tag{52}$$

The resulting classification function is

$$g(x) = w_\Phi \cdot \Phi(x) + b_\Phi. \tag{53}$$

It is not a problem that Φx is not known explicitly, since

$$w_\Phi \cdot \Phi x = \sum_{i=1}^{n} \alpha_i k(x_i, x). \tag{54}$$

b_Φ is the threshold in the kernelized feature space

$$b_\Phi = -w_\Phi \cdot m_\Phi \tag{55}$$

where m_Φ is the midpoint of the two class means in the feature space:

$$m_\Phi = \frac{1}{2}(\mu_1^\Phi + \mu_2^\Phi) = \frac{1}{2}\left(\frac{1}{n_1}\sum_{i=1}^{n_1} \Phi(x_i^{(1)}) + \frac{1}{n_2}\sum_{i=1}^{n_2} \Phi(x_i^{(2)})\right) \tag{56}$$

The resulting threshold is

$$b_\Phi = -\frac{1}{2}\left(\frac{1}{n_1}\sum_{i=1}^{n_1}\sum_{j=1}^{n}\alpha_j k(x_i^{(1)}, x_j) + \frac{1}{n_2}\sum_{i=1}^{n_2}\sum_{j=1}^{n}\alpha_j k(x_i^{(2)}, x_j)\right) \tag{57}$$

where $x_i^{(1)}$ is the i^{th} sample of class 1 (or 2). The final kernelized LDA classifier is

$$g(x) = b_\Phi + \sum_{i=1}^{n}\alpha_i k(x_i, x). \tag{58}$$

APPENDIX C

FDA MICROARRAY QUALITY CONTROL PHASE II (MAQC-II) PROJECT

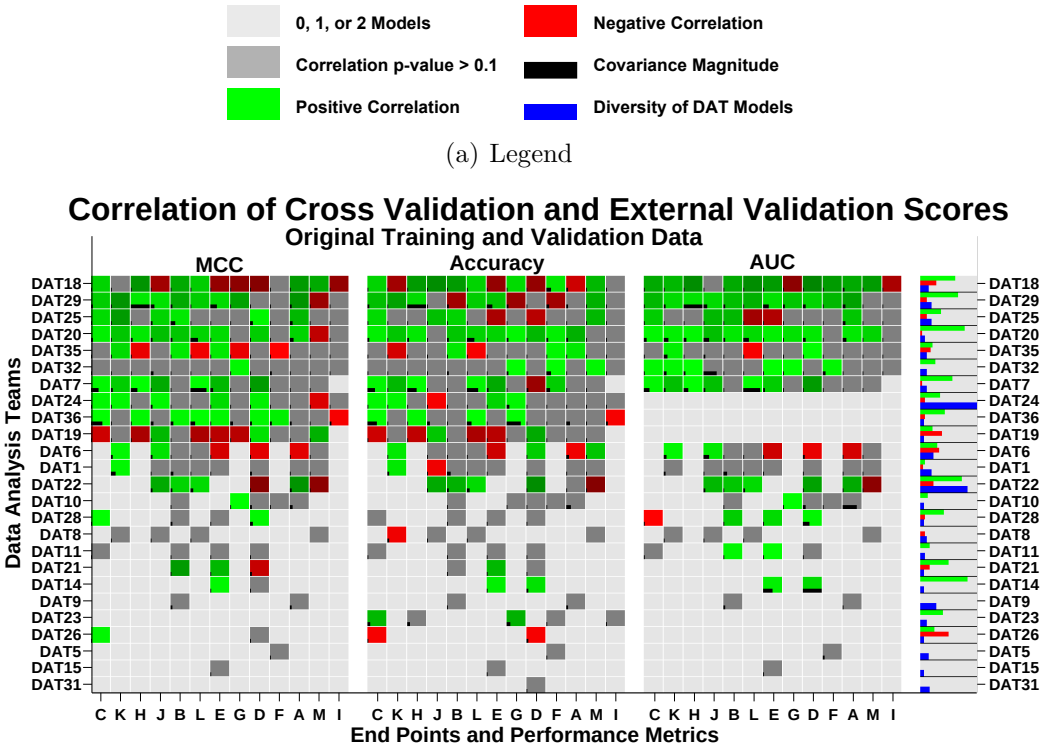
The FDA recently conducted a large-scale study on the feasibility of using microarray data for clinical prediction. In a collaboration of 36 data analysis teams (DAT), the Microarray Quality Control Consortium (MAQC) analyzed six large microarray datasets that covered 13 clinical and toxicity endpoints. Each DAT was free to construct a data analysis protocol (DAP) within certain guidelines. These guidelines required that all predictive models be assessed using full cross validation (10 iterations of 5-fold cross validation) and that each DAT select a single candidate model for testing with a blind validation dataset. In general, the prediction models submitted by the teams were diverse, resulting in a wide variety of prediction performances. Overall correlation of internal cross validation and external validation scores was reasonable. However, within each endpoint and for each DAT, concordance of internal cross validation and external validation was variable. DATs were also required to swap the training and testing data to determine if the performance of their predictors could be replicated when sample populations change.

Figure 64 and **Figure 65** are detailed summaries of the correlation of internal and external validation for each DAT and endpoint for both the blind (**Figure 64**) and swapped (**Figure 65**) experiments. Summarizing the positive (green) and negative (red) correlations reveals that AUC performs better than MCC and accuracy in terms of fraction of positive correlations (**Figure 64(d)** and **Figure 65(d)**).

Although correlations are computed using only models within each DAT and end point pair, we must still consider the diversity of models within in each subgroup.

A DAT with a diverse set of models may artificially inflate the correlation if, for example, the range of performance scores is large. As such, we compute the relative diversity of each DAT (blue horizontal bars on the right side of **Figure 64(b)** and **Figure 65(b)**). The diversity is the number of unique feature selection and classifier pairs within the total set of models for each DAT. Examining **Figure 64(b)** closely, we see that DAT24 and DAT22 have the highest model diversity. However, this does not lead to predominately positive correlations. Similarly, DAT29 and DAT20 have a relatively less diverse set of models, yet produced more positive correlations. We also examine the absolute covariance of cross validation and external validation, since the variance of performance scores may also inflate the correlation coefficient. Absolute covariance for each DAT and end point pair is represented as a small black bar, the length of which is proportional to absolute covariance. Generally, a large absolute variance corresponds to a significantly positive correlation. However, some cases, such as for DATs in endpoint A, are negatively correlated or are not significantly correlated but still have a high absolute covariance. Thus, a large range of performance metric scores may usually lead to a positive correlation, but not in all cases. **Figure 64(c)** and **Figure 65(c)** are summaries of positive and negative correlation for each end point, ordered by decreasing positive minus negative correlation. Endpoints M and I, both of which are the result of random class label assignment, have the worst performance in terms of correlating cross validation and external validation. The order of the end points roughly follows the order of increasing prediction difficulty.

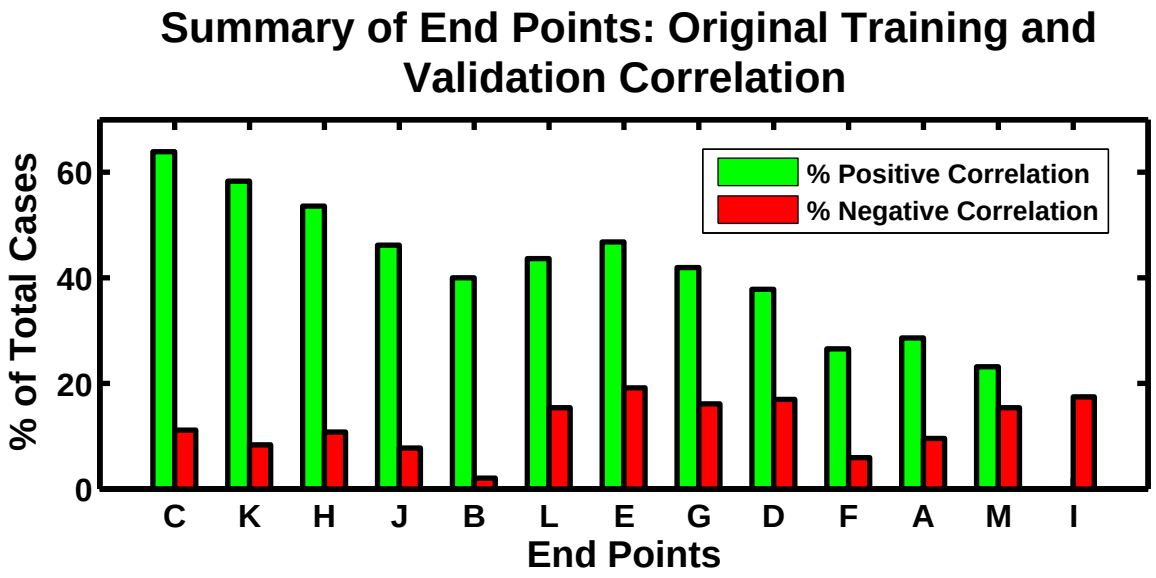
Data analysis teams in the MAQC consortium selected candidate models using different classification performance metrics. In **Figure 66**, we investigate the effect of the performance metric on candidate model selection using cross validation results from 36 data analysis teams and 13 data endpoints. Intuitively, one would assume that that a classification model that performs well according to some metric 'A' should also perform well according to some other metric 'B'. However, according to **Figure**



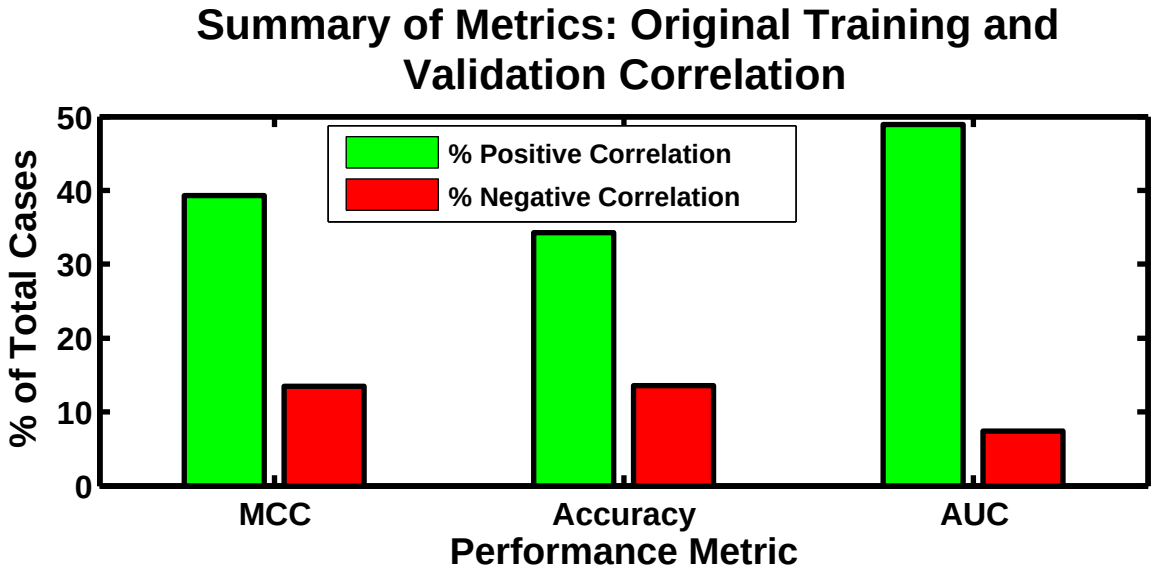
(a) Legend

(b) Correlation between predictive model internal cross validation and external blind validation.

Figure 64: Correlation of predictive model internal cross validation performance to external blind validation performance. **64(b):** DATs computed classification model performance using three performance metrics (MCC, accuracy, and AUC) averaged over 10 iterations of 5-fold cross validation. At least three models from both internal cross validation and external validation are required to compute correlation for each DAT and end point pair. Light gray squares indicate that only zero, one, or two models are available. DATs that have not provided enough data to compute correlation for any end point have been excluded. Green squares indicate a positive correlation between internal cross validation scores and external validation scores. Red squares indicate negative correlation. The brightness of red and green squares indicates the degree of correlation, i.e., a larger absolute Pearson's correlation coefficient results in a lighter square. Dark gray squares indicate that the p-value of correlation is larger than 0.1. The black bar within each box represents the absolute covariance. Data analysis teams are sorted from top to bottom by decreasing number of endpoints analyzed, then by decreasing total number of models. Endpoints are sorted from left to right by increasing percentage of positive correlations minus negative correlations. The image bar on the right summarizes each DAT with the percentage of positive correlations (green), negative correlations (red), and relative diversity of the DAT (blue). Diversity is a measure of the number of unique feature selection/classification methods used. **64(c)** (next page): Summary of the positive and negative correlations for each end point. **64(d)** (next page): Summary of the positive and negative correlations for each performance metric.



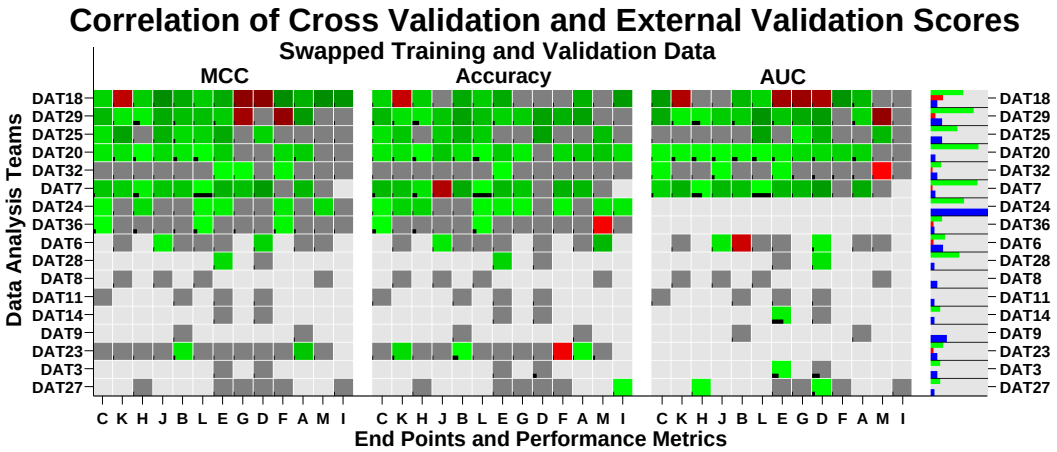
(c) Summary of Endpoints: Correlation of internal cross validation to external blind validation.



(d) Summary of Performance Metrics: Correlation of internal cross validation to external blind validation.

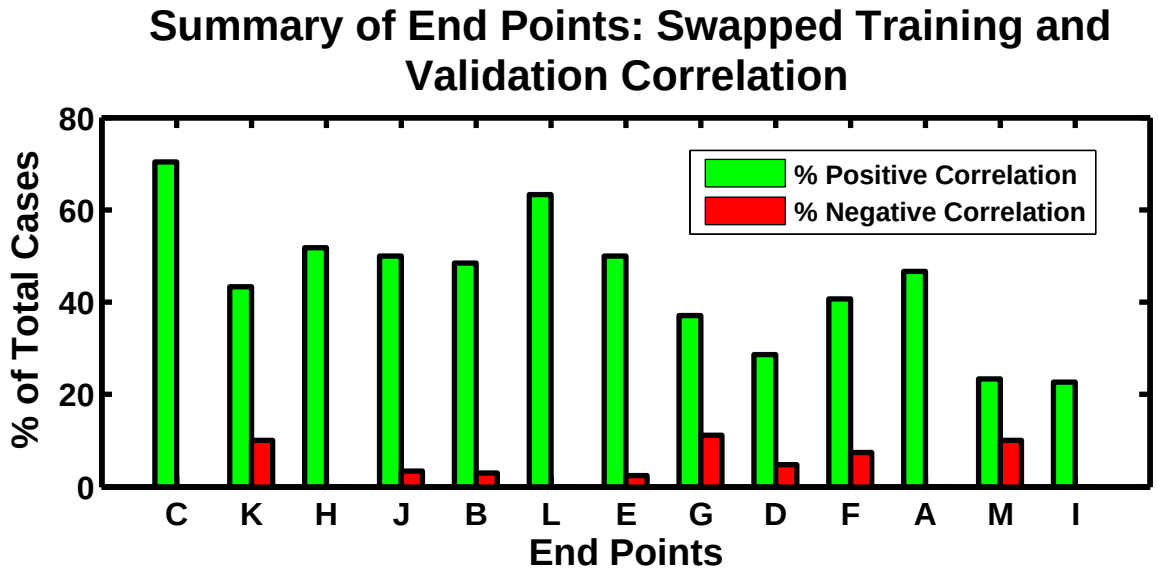
Figure 64 parts (c) and (d). Figure parts (a) and (b) and full caption on the previous page.

(a) Legend

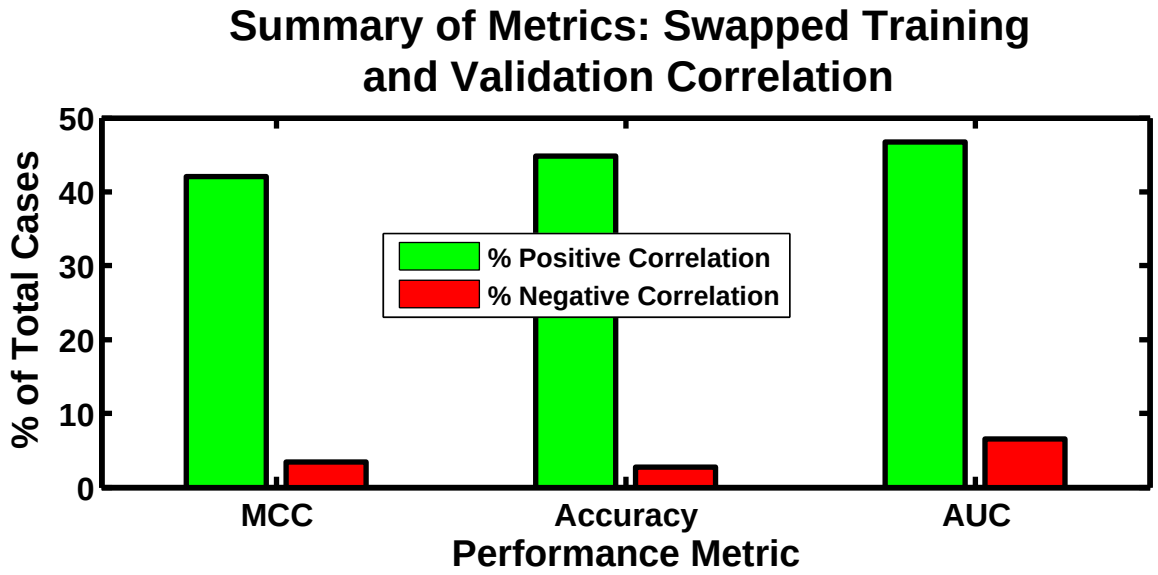


(b) Correlation between predictive model internal cross validation and external validation after swapping the training and testing data.

Figure 65: Correlation of predictive model internal cross validation performance to external validation performance after swapping training and testing data. **65(b):** DATs computed classification model performance using three performance metrics (MCC, accuracy, and AUC) averaged over 10 iterations of 5-fold cross validation. At least three models from both internal cross validation and external validation are required to compute correlation for each DAT and end point pair. Light gray squares indicate that only zero, one, or two models are available. DATs that have not provided enough data to compute correlation for any end point have been excluded. Green squares indicate a positive correlation between internal cross validation scores and external validation scores. Red squares indicate negative correlation. The brightness of red and green squares indicates the degree of correlation, i.e., a larger absolute Pearson's correlation coefficient results in a lighter square. Dark gray squares indicate that the p-value of correlation is larger than 0.1. The black bar within each box represents the absolute covariance. Data analysis teams are sorted from top to bottom by decreasing number of endpoints analyzed, then by decreasing total number of models. Endpoints are sorted from left to right by increasing percentage of positive correlations minus negative correlations (using the original, not swapped, data). The image bar on the right summarizes each DAT with the percentage of positive correlations (green), negative correlations (red), and relative diversity of the DAT (blue). Diversity is a measure of the number of unique feature selection/classification methods used. **65(c)** (next page): Summary of the positive and negative correlations for each end point. **65(d)** (next page): Summary of the positive and negative correlations for each performance metric.

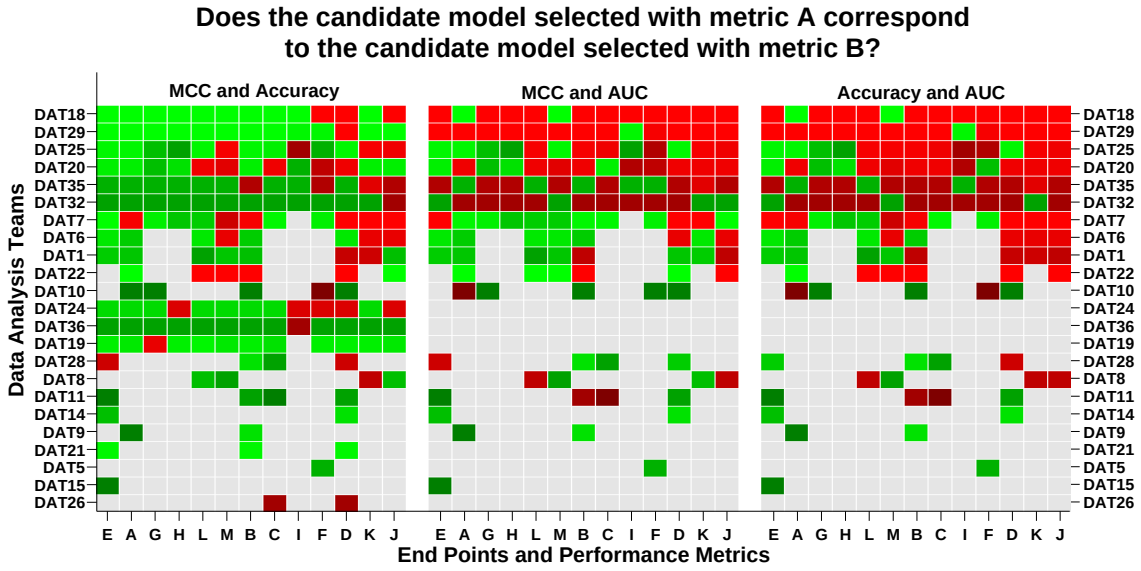


(c) Summary of Endpoints: Correlation of internal cross validation to external validation after swapping the training and testing data.



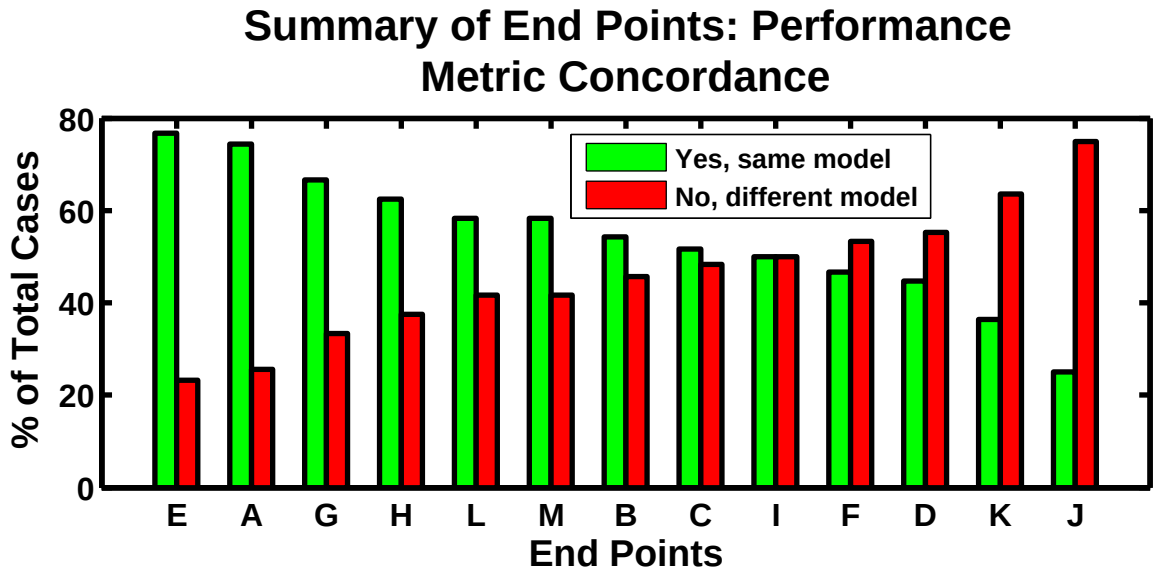
(d) Summary of Performance Metrics: Correlation of internal cross validation to external validation after swapping the training and testing data.

Figure 65 parts (c) and (d). Figure parts (a) and (b) and full caption on the previous page.

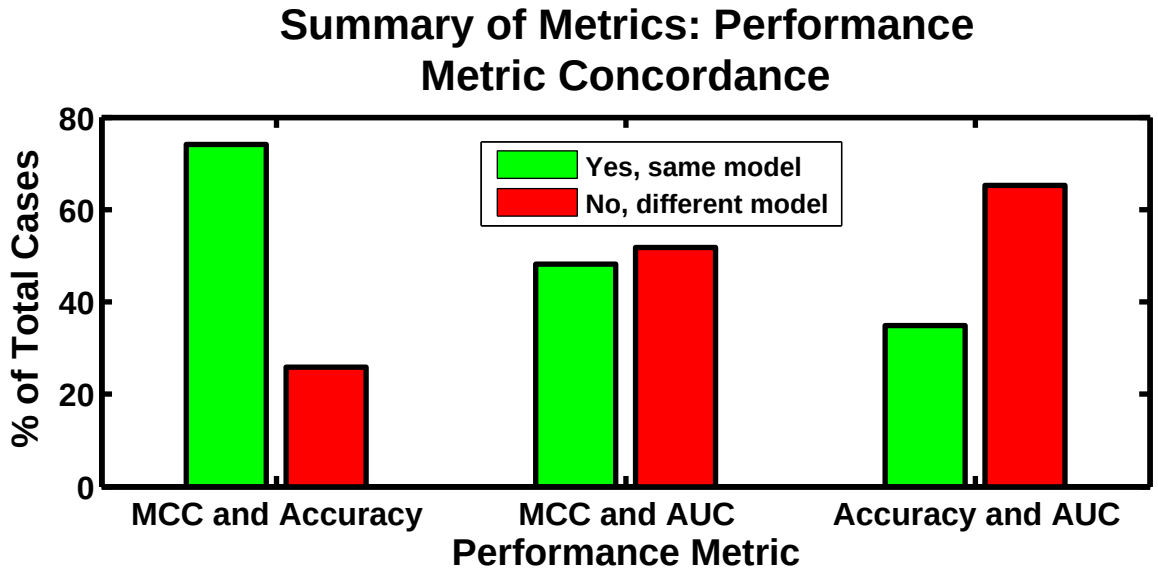


(a) Concordance of candidate model selection using different performance metrics.

Figure 66: Concordance of candidate model selection using different performance metrics. **66(a):** Each cell represents a measure of model selection concordance using one of three pairs of performance metric combinations: (MCC and Accuracy, left column), (MCC and AUC, middle column), and (Accuracy and AUC, right column). Classification model performance was computed using these performance metrics and averaging 10 iterations of 5-fold cross validation. Light gray squares indicate that there were less than three data points available. Green squares indicate that the best model selected using metric A is the same as the best model selected using metric B. Red indicates that the best models were not the same. Brightness of the red and green squares indicate the significance of the comparison. A brighter square means that there were more models to choose from. **66(b)** (next page): Summary of the percentage of model matches for each endpoint, ordered by percentage of matches minus percentage of mismatches. **66(c)** (next page): Summary of the percentage of model matches for each performance metric comparison.



(b) Summary of Endpoints: Concordance of candidate model selection using different performance metrics.



(c) Summary of Performance Metrics: Concordance of candidate model selection using different performance metrics.

Figure 66 parts (b) and (c). Figure part (a) and full caption on the previous page.

66, this is sometimes not true. This result correlates to those of a previous study that examined several performance metrics and concluded that each metric measures a different aspect of performance [40]. In the MAQC empirical study, we can see that there are varying degrees of concordance between selected candidate models depending on which performance metrics we compare. For example, there is a high concordance between the MCC and accuracy metrics (**Figure 66(a)**, left panel). Comparing AUC to either MCC or accuracy, however, reveals that AUC is quite different (**Figure 66(a)**, middle and right panels). In fact, the AUC measures the correctness of rank order whereas the MCC and accuracy measure only performance of binary classification. **Figure 66(c)** summarizes the level of concordance between each pair of performance metrics. MCC and accuracy agree in approximately 70% of cases, MCC and AUC agree in 50%, and AUC and accuracy agree in less than 40%.

APPENDIX D

CLASSIFICATION PERFORMANCE METRICS

When building predictive models for clinical use, we often want to assess classification performance in order to tune model parameters. Some of these model parameters include feature size and various classifier properties. We consider only the case of two classes, i.e., the classifier attempts to label samples as either positive or negative depending on sample features. We assume that samples cannot simultaneously belong to both positive and negative groups. Consequently, we can define four numbers that describe the resulting classification. TP, the true positive count, is the number of samples classified as positive that are actually positive. TN, the true negative count, is the number of samples classified as negative that are actually negative. FP, the false positive count, is the number of samples classified as positive that are actually supposed to be negative. And FN, the false negative count, is the number of samples classified as negative that are actually supposed to be positive. We can summarize these four variables in the following table:

Table 26: Summarizing the performance of a prediction rule.

	Actual Positives (AP)	Actual Negatives (AN)	Total
Predicted Positives (PP)	TP	FP	P'
Predicted Negatives (PN)	FN	TN	N'
Total	P	N	

D.1 Accuracy

The simplest performance metric is accuracy, which is the ratio of correctly classified samples to the total number of samples:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{59}$$

We can also represent this information as fractions of the total number of samples. For example, the true positive fraction is the ratio of the total number of true positives to the actual number of positive samples:

$$TPF = \frac{TP}{TP + FN} \tag{60}$$

Similarly, the false positive fraction is the ratio of the total number of false positives to the actual number of negative samples:

$$FPF = \frac{FP}{FP + TN} \tag{61}$$

TPF is also known as **sensitivity** (Se), which is the rate at which a test correctly identifies positive samples. In terms of disease classification, if the disease state is the positive class, it is the rate at which a test correctly identifies diseased individuals as such. **Specificity** (Sp) is the rate at which a test correctly identifies negative samples. Again, in terms of disease classification, it is the rate at which a test correctly identifies the disease-free individuals as such. Specificity is equivalent to $1 - FPF$.

In theory, a random classifier should yield 50% accuracy. In practice, however, training and validation data rarely have equal class sample sizes. Consequently, the measure of classification accuracy may be misleading. For example, assume we have 80 positive samples and 20 negative samples with sensitivity (TPF) and specificity (1-FPF) fixed at 0.75 and 0.0, respectively. Accuracy for this case would be (0.75*80+0.0*20)/(80+20)=0.6. However, if we have 50 positive samples and 50 negative samples, then the accuracy would be (0.75*50+0.0*50)/(50+50)=0.375. Thus,

accuracy is dependent on prevalence. We can show this more generally by defining N+ and N- as the number of positive and negative samples, respectively, and computing the accuracy as

$$Accuracy = \frac{Se \cdot N^+ + Sp \cdot N^-}{N^+ + N^-}. \tag{62}$$

If we define prevalence as the ratio of positive samples to the total number of samples

$$P = \frac{N^+}{N^+ + N^-}, \tag{63}$$

then we may also represent accuracy as

$$Accuracy = Se \cdot P + Sp \cdot (1 - P), \tag{64}$$

indicating that accuracy is a weighted average of sensitivity and specificity. We can directly observe the effect of prevalence on accuracy by plotting accuracy as a function of TPF, FPF, and P:

$$Accruacy = F_{Acc}(TPF, FPF, P) = TPF \cdot P + (1 - FPF) \cdot (1 - P). \tag{65}$$

For each value of P, accuracy is a plane that tilts on the $TPF = 1 - FPF = Acc$ line (**Figure 67**). Accuracy values for fixed TPF and FPF vary significantly when TPF and FPF are either both large or both small. We can explain this in another way by observing that when either sensitivity or specificity is close to zero, accuracy heavily depends on prevalence.

D.2 Area Under the ROC Curve (AUC)

While accuracy measures the performance of a classifier using a single threshold, the receiver/operating characteristic (ROC) curve captures the performance at all possible thresholds. For example, most classifiers will assign continuous numeric values to each sample, which are then used to make class assignments based on threshold. For a particular threshold, we can compute the TPF and FPF values. By varying the threshold, we can observe multiple TPF and FPF pairs. These

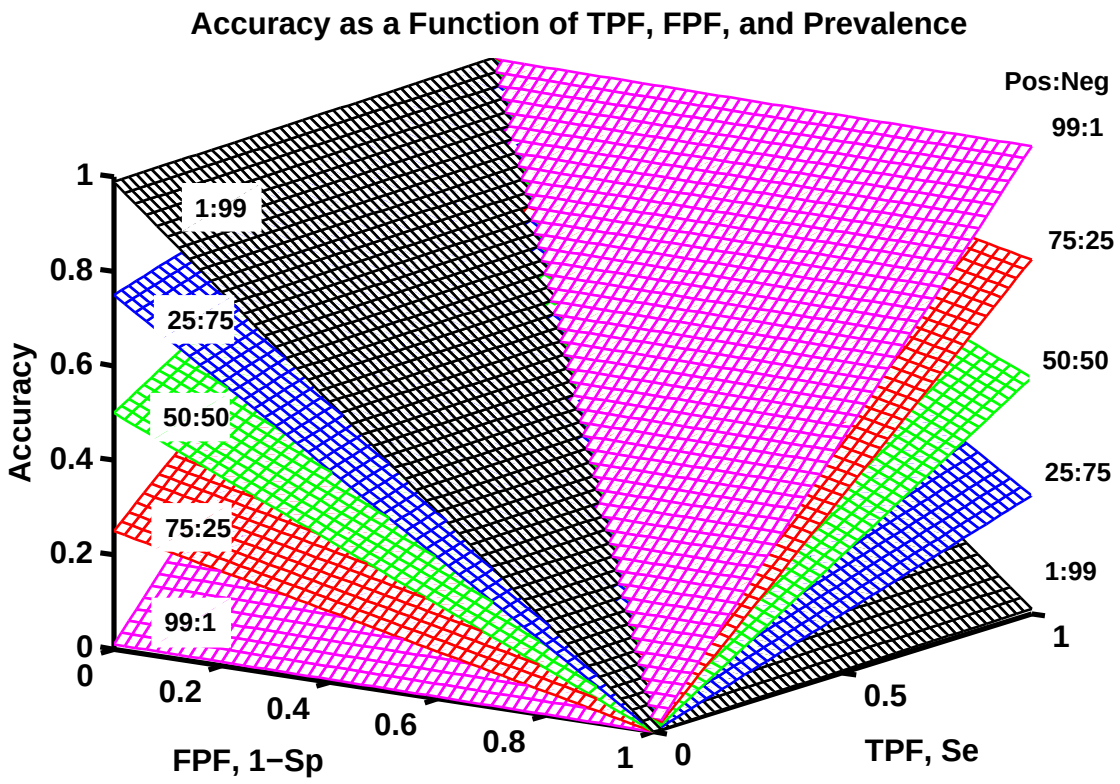


Figure 67: Accuracy as a function of TPF, FPF, and Prevalence. Each surface corresponds to a different prevalence ratio.

pairs trace a curve that ranges from $TPF = FPF = 0$ to $TPF = FPF = 1$. If all thresholds are perfect (i.e., $FP = 0$ and $FN = 0$ for all thresholds), then the ROC curve rises along the vertical line $FPF = 0$ and runs along the horizontal line $TPF = 1$. If the classifier performs no better than random chance, the ROC curve runs along the line $FPF = TPF$. Thus, we can measure the overall performance of the classifier at all possible thresholds by integrating the area under the ROC curve, or AUC [8, 76]. We can compute AUC using the formula

$$AUC = \frac{1}{N^+ \cdot N^-} \left(\sum_{i=1}^{N^+} \sum_{j=1}^{N^-} I(x_i > y_j) + \frac{1}{2} \sum_{i=1}^{N^+} \sum_{j=1}^{N^-} I(x_i = y_j) \right) \tag{66}$$

where x_i and y_j are continuous classifier outputs for positive and negative samples, respectively. Note that in the case of ties, the summation is weighted by 0.5.

If a classifier returns only binary values, AUC reduces to the expected value at a single pair of TPF and FPF. The binary AUC is the average of sensitivity and specificity

$$AUC_{bin} = F_{AUC}(TPF, FPF) = \frac{1}{2}(Se + Sp) = \frac{1}{2}(TPF - FPF + 1), \tag{67}$$

which is similar to the expression for accuracy except it is independent of prevalence, P.

D.3 Matthews Correlation Coefficient (MCC)

Although AUC is attractive because of its invariance to prevalence, it measures the performance of a classifier at all thresholds, most of which may not be relevant to the problem at hand. The Matthews Correlation Coefficient (MCC) attempts to find a compromise between accuracy and AUC. The formula for MCC is a non-linear function of TP, TN, FP, and FN:

$$\begin{aligned} MCC &= F_{MCC}(TPF, FPF, P) \\ &= \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \end{aligned} \tag{68}$$

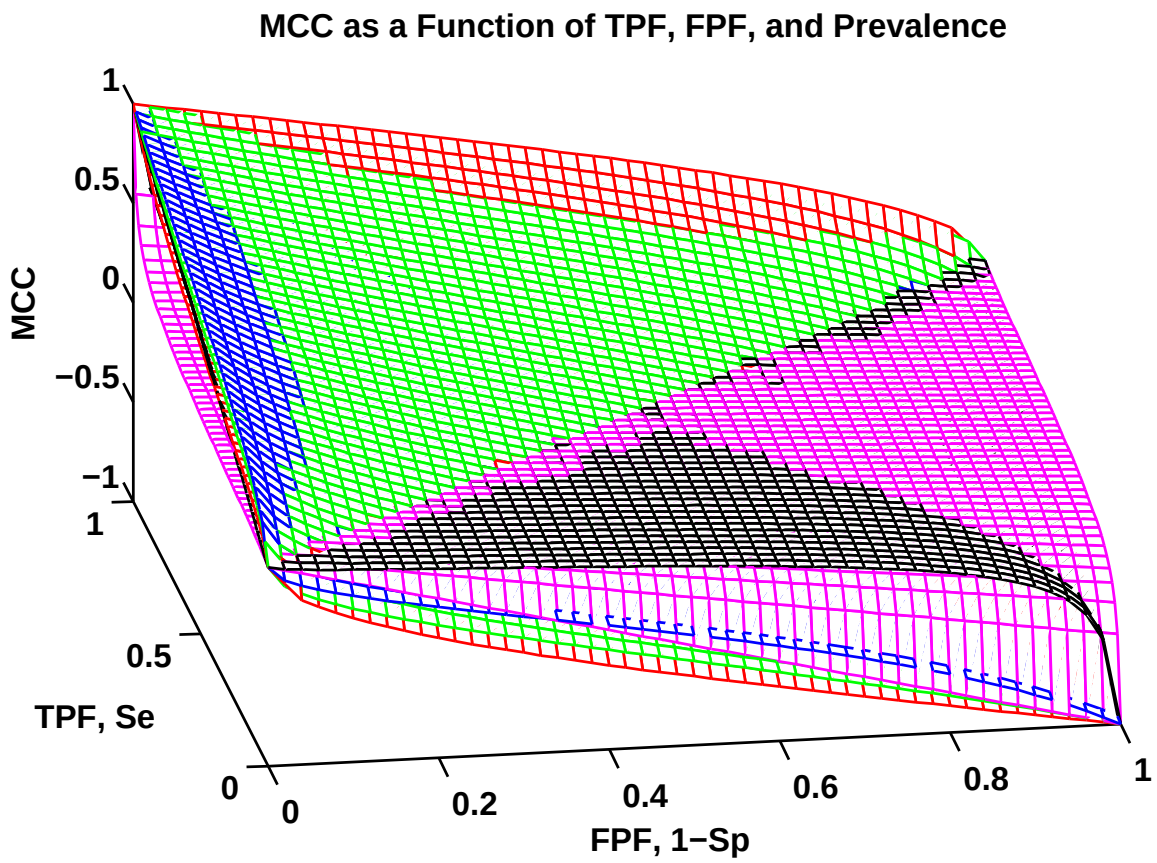


Figure 68: MCC as a function of TPF, FPF, and Prevalence. Each prevalence corresponds to a different prevalence ratio.

In **Figure 68**, we can see that as prevalence changes, the MCC surfaces still change, but remain relatively close to the binary AUC surface compared to accuracy.

APPENDIX E

SELECTED PUBLICATIONS

The work presented in this thesis is a culmination of several years of research that resulted in the following peer reviewed journal publications, book chapters, and conference proceedings.

Journals

1. **Phan JH**, Moffitt RA, Stokes TH, Liu J, Young AN, Nie S, and Wang MD. Convergence of Biomarkers, Bioinformatics, and Nanotechnology for Individualized Cancer Treatment. *Trends Biotechnol.* In Press. 2009.

2. The MicroArray Quality Control (MAQC) Consortium (**contributing author**). The MAQC-II Project: A comprehensive study of common practices for the development and validation of microarray-based predictive models. Submitted to *Nat Biotechnol.* 2009.

3. Parry RM, Jones W, Stokes TH, **Phan JH**, Moffitt RA, Fang H, Shi L, Oberthuer A, Fischer M, Tong W, Wang MD. K-nearest neighbors (KNN) models for microarray gene-expression analysis and reliable clinical outcome prediction. Submitted to *Nat Biotechnol.* 2009.

4. Jones W, Wang MD, Moffitt RA, **Phan JH**, Stokes TH, Bao W, Wolfinger R, Li L, Parker J. The impact of quality-related artifacts on the predictive performance of Affymetrix GeneChips: a robustness case study within MAQC-II. Submitted to *Nat Biotechnol.* 2009.

5. Stokes TH, Moffitt RA, **Phan JH**, and Wang MD. chip artifact CORRECTion (caCORRECT): a bioinformatics system for quality assurance of genomics and proteomics array data. *Ann Biomed Eng.* 35(6):1068-1080. 2007.

6. Yin-Goen Q, Dale J, Yang WL, **Phan JH**, Moffitt RA, Petros JA, Datta MW, Amin MB, Wang MD, and Young AN. Advances in molecular classification of renal neoplasms. *Histology and Histopathology.* 21:325-339. 2006.

Book Chapters

1. **Phan JH**, Quo CF, and Wang MD. Functional genomics and proteomics in the clinical neurosciences: data mining and bioinformatics. *Prog Brain Res.* 158:83-108. 2006.

Conference Proceedings

1. **Phan JH**, Yin-Goen Q, Young AN, and Wang MD. Improving the Efficiency of Biomarker Identification Using Biological Knowledge. *Pac Symp Biocomput.* 14:427-438. 2009.

2. **Phan JH**, Moffitt RA, Barrett AB, and Wang MD. Improving Microarray Sample Size Using Bootstrap Data Combination. *Proceedings of the International Multi-Symposiums on Computer and Computational Sciences, IMSCCS.* 2008:37-44. 2008.

3. **Phan JH** and Wang MD. Estimating Classification Error to Identify Biomarkers in Time Series Expression Data. *Proceedings of the 7th IEEE International Conference on BioInformatics and BioEngineering, BIBE.* 2007:172-179. 2007.

4. **Phan JH**, Young AN, and Wang MD. Selecting Clinically-Driven Biomarkers for Cancer Nanotechnology. *Proceedings of the 28th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS.* 2006:3317-3320. 2006.

REFERENCES

[1] Aerts, S., Lambrechts, D., Maity, S., Van Loo, P., Coessens, B., De Smet, F., Tranchevent, L., De Moor, B., Marynen, P., Hassan, B., Carmeliet, P., and Moreau, Y., "Gene prioritization through genomic data fusion," *Nat Biotechnol*, vol. 24, no. 5, pp. 537–544, 2006.

[2] Ambroise, C. and McLachlan, G., "Selection bias in gene extraction on the basis of microarray gene-expression data," *PNAS*, vol. 99, no. 10, pp. 6562–6566, 2002.

[3] Bagchi, A., Papazoglu, C., Wu, Y., Capurso, D., Brodt, M., Francis, D., Bredel, M., Vogel, H., and Mills, A., "CHD5 Is a Tumor Suppressor at Human 1p36," *Cell*, vol. 128, no. 3, pp. 459–475, 2007.

[4] Beissbarth, T. and Speed, T., "GOstat: Find statistically overrepresented Gene Ontologies within a group of genes," *Bioinformatics*, vol. 20, no. 9, pp. 1464–1465, 2004.

[5] Bellazzi, R. and Zupan, B., "Towards knowledge-based gene expression data mining," *Journal of Biomedical Informatics*, vol. 40, no. 6, pp. 787–802, 2007.

[6] Boczko, E. M. and Young, T. R., "The Signed Distance Function: A New Tool for Binary Classification," 2005.

[7] Bolstad, B., Irizarry, R., Astrand, M., and Speed, T., "A Comparison of Normalization Methods for High Density Oligonucleotide Array Data Based on Bias and Variance," *Bioinformatics*, vol. 19, pp. 185–193, 2003.

[8] Bradley, A., "The use of the area under the ROC curve in the evaluation of machine learning algorithms," *Pattern Recognition*, vol. 30, no. 7, pp. 1145–1159, 1997.

[9] Braga-Neto, U. and Dougherty, E., "Bolstered error estimation," *Pattern Recognition*, vol. 37, pp. 1267–1281, 2004.

[10] Braga-Neto, U. and Dougherty, E., "Is cross-validation valid for small-sample microarray classification?," *Bioinformatics*, vol. 20, pp. 374–380, August 2004.

[11] Brazma, A., Hingamp, P., Quackenbush, J., Sherlock, G., Spellman, P., Stoeckert, C., Aach, J., Ansorge, W., Ball, C., Causton, H., Gaasterland, T., Glenisson, P., Holstege, F., Kim, I., Markowitz,

V., Matese, J., Parkinson, H., Robinson, A., Sarkans, U., Schulze-Kremer, S., Stewart, J., Taylor, R., Vilo, J., and Vingron, M., "Minimum information about a microarray experiment (MIAME)-toward standards for microarray data," *Nature Genetics*, vol. 29, pp. 365–371, 2001.

[12] Brodsky, L., Leontovich, A., Shtutman, M., and Feinstein.E., "Identification and handling of artifactual gene expression profiles emerging in microarray hybridization experiments," *Nucleic Acids Research*, vol. 32, no. 4, p. e46, 2004.

[13] Bronzetti, E., Artico, M., Forte, F., Pagliarella, G., Felici, L., D'Ambrosio, A., Vespasiani, G., and Bronzetti, B., "A possible role of BDNF in prostate cancer detection," *Oncol Rep*, vol. 19, no. 4, pp. 969–74, 2008.

[14] Buness, A., Huber, W., Steiner, K., Sultmann, H., and Poustka, A., "arrayMagic: two-colour cDNA microarray quality control and preprocessing," *Bioinformatics*, vol. 21, no. 4, pp. 554–556, 2005.

[15] Burcombe, R., Wilson, G., Dowsett, M., Khan, I., Richman, P., Daley, F., Detre, S., and Makris, A., "Evaluation of Ki-67 proliferation and apoptotic index before, during and after neoadjuvant chemotherapy for primary breast cancer," *Breast Cancer Research*, vol. 8, no. 3, p. R31, 2006.

[16] Buyse, M., Loi, S., van't Veer, L., Viale, G., Delorenzi, M., Glas, A., Saghatchian d'Assignies, M., Bergh, J., Lidereau, R., and Ellis, P., "Validation and Clinical Utility of a 70-Gene Prognostic Signature for Women With Node-Negative Breast Cancer," *JNCI*, vol. 98, no. 17, pp. 1183–1192, 2006.

[17] Canales, R., Luo, Y., Willey, J., Austermiller, B., Barbacioru, C., Boysen, C., Hunkapiller, K., Jensen, R., Knight, C., Lee, K., Ma, Y., Maqsodi, B., Papallo, A., Peters, E., Poulter, K., Ruppel, P., Samaha, R., Shi, L., Yang, W., Zhang, L., and Goodsaid, F., "Evaluation of DNA microarray results with quantitative gene expression platforms," *Nature Biotechnology*, vol. 24, pp. 1115–1122, 2006.

[18] Carbon, S., Ireland, A., Mungall, C., Shu, S., Marchall, B., Lewis, S., AmiGO Hub, and Web Presence Working Group, "AmiGO: online access to ontology and annotation data," *Bioinformatics*, vol. 25, no. 2, pp. 288–289, 2009.

[19] Chandran, U., Dhir, R., Ma, C., Michalopoulos, G., Becich, M., and Gilbertson, J., "Differences in gene expression in prostate cancer, normal appearing prostate tissue adjacent to cancer and prostate tissues from cancer free organ donors," *BMC Cancer*, vol. 5, 2005.

[20] Chang, C.-C. and Lin, C.-J., *LIBSVM: a library for support vector machines*, 2001. Software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.

[21] Chen, L., Xuan, J., Wang, C., Shih, I., Wang, Y., Zhang, Z., Hoffman, E., and Clarke, R., "Knowledge-guided multi-scale independent component analysis for biomarker identification," *BMC Bioinformatics*, vol. 9, no. 416, 2008.

[22] Chen, Y., Tu, J., Kao, J., Zhou, X., and Mazumdar, M., "Messenger RNA Expression Ratios among Four Genes Predict Subtypes of Renal Cell Carcinoma and Distinguish Oncocytoma and Carcinoma," *Clin Cancer Res*, vol. 11, no. 18, pp. 6558–6566, 2005.

[23] Chin, S., Wang, Y., Thorne, N., Teschendorff, A., Pinder, S., Vias, M., Naderi, A., Roberts, I., Barbosa-Morais, N., Garcia, M., Iyer, N., Kranjac, T., Robertson, J., Aparicio, S., Tavare, S., Ellis, I., Brenton, J., and Caldas, C., "Using array-comparative genomic hybridization to define molecular portraits of primary breast cancers," *Oncogene*, vol. 26, no. 13, pp. 1959–1970, 2007.

[24] Choi, J., Yu, U., Kim, S., and Yoo, O., "Combining multiple microarray studies and modeling insterstudy variation," *Bioinformatics*, vol. 19, pp. i84–i90, 2003.

[25] Chuaqui, R., Bonner, R., Best, C., Gillespie, J., Flaig, M., Hewitt, S., Phillips, J., Krizman, D., Tangrea, M., Ahram, M., Linehan, W., Knezevic, V., and Emmert-Buck, M., "Post-analysis follow-up and validation of microarray experiments," *Nature Genetics*, vol. 32, pp. 509–514, 2002.

[26] Cimino, J., Hayamizu, T., Bodenreider, O., Davis, B., Stafford, G., and Ringwald, M., "The caBIG terminology review process," *Journal of Biomedical Informatics*, vol. Epub, 2008.

[27] Coons, A., Creech, H., and Jones, R., "Immunological properties of an antibody containing a fluorescent group," *Proc Soc Exp Biol Med*, vol. 47, pp. 200–202, 1941.

[28] Cristianini, N. and Shawe-Taylor, J., *An Introduction to Support Vector Machines and other Kernel-based Learning Methods*. Cambridge University Press, 2000.

[29] de Reynies, A., Assie, G., Rickman, D., Tissier, F., Groussin, L., Rene-Corail, F., Dousset, B., Bertagna, X., Clauser, E., and Bertherat, J., "Gene Expression Profiling Reveals a New Classification of Adrenocortical Tumors and Identifies Molecular Predictors of Malignancy and Survival," *J Clin Oncol*, vol. 19, no. 7, pp. 1108–1115, 2009.

[30] de Souto, M., Costa, I., de Araujo, D., Ludermir, T., and Schliep, A., "Clustering cancer gene expression data: a comparative study," *BMC Bioinformatics*, vol. 9, no. 497, 2008.

[31] Ding, C. and Peng, H., "Minimum Redundancy Feature Selection From Microarray Gene Expression Data," *Journal Bioinformatics and Computational Biology*, vol. 3, no. 2, pp. 185–205, 2005.

[32] Duda, R., Hart, P., and Stork, D., *Pattern Classification*. Wiley-Interscience, 2000.

[33] Dudoit, S., Fridlyand, J., and Speed, T., "Comparison of Discrimination Methods for the Classification of Tumors Using Gene Expression Data," *Journal of the American Statistical Association*, vol. 97, no. 457, pp. 77–87, 2002.

[34] Eble, J., Sauter, G., Epstein, J., and Sesterhenn, I., *Tumors of the kidney*. IARC Press, 2004.

[35] Edgar, R., Domrachev, M., and Lash, A., "Gene Expression Omnibus: NCBI gene expression and hybridization array data repository," *Nucleic Acids Research*, vol. 30, no. 1, pp. 207–210, 2002.

[36] Efron, B., "Estimating the Error Rate of a Prediction Rule: Improvement on Cross-Validation," *J Amer Statistical Assoc*, vol. 78, no. 382, pp. 316–331, 1983.

[37] Efron, B. and Tibshirani, R., "Improvements on Cross-Validation: The .632+ Bootstrap Method," *J Amer Statistical Assoc*, vol. 92, no. 438, pp. 548–560, 1997.

[38] Eisen, M., Spellman, P., Brown, P., and Botstein, D., "Cluster analysis and display of genome-wide expression patterns," *PNAS*, vol. 95, pp. 14863–14868, 1998.

[39] Ernst, T., Hergenhahn, M., Kenzelmann, M., Cohen, C., Bonrouhi, M., Weninger, A., Klaren, R., Grone, E., Wiesel, M., Gudemann, C., Kuster, J., Schott, W., Staehler, G., Kretzler, M., Hollstein, M., and Grone, H.-J., "Decrease and Gain of Gene Expression Are Equally Discriminatory Markers for Prostate Carcinoma: A Gene Expression Analysis on Total and Microdissected Prostate Tissue," *Am J Path*, vol. 160, no. 6, pp. 2169–2180, 2002.

[40] Ferri, C., Hernandez-Orallo, J., and Modroiu, R., "An experimental comparison of performance measures for classification," *Pattern Recognition Letters*, vol. 30, pp. 27–38, 2009.

[41] Fogel, G., "Computational intelligence approaches for pattern discovery in biological systems," *Briefings in Bioinformatics*, vol. 2, no. 4, pp. 307–316, 2008.

[42] Freimuth, R., Shauer, M., Lodha, P., Govindrao, P., Nagarajan, R., and Chute, C., "caBIG compatibility review system: software to support the evaluation of applications using defined interoperability criteria," *AMIA Annu Symp Proc*, pp. 197–201, 2008.

[43] Frijters, R., Heupers, B., van Beek, P., Bouwhuis, M., van Schaik, R., de Vlieg, J., Polman, J., and Alkema, W., "CoPub: a literature-based keyword enrichment tool for microarray data analysis," *Nucleic Acids Research*, vol. 36, no. Web Server, pp. W406–W410, 2008.

[44] Fu, W., Carroll, R., and Wang, S., "Estimating misclassification error with small samples via bootstrap cross-validation," *Bioinformatics*, vol. 21, pp. 1979–1986, 2005.

[45] Gabriele, L., Moretti, F., Pierotti, M., Marincola, F., Foa, R., and Belardelli, F., "The use of microarray technologies in clinical oncology," *J Trans Med*, vol. 4, 2006.

[46] Gene Ontology Consortium, "Gene Ontology: tool for the unification of biology," *Nature Genetics*, vol. 25, pp. 25–29, 2000.

[47] Gentleman, R., Carey, V., Bates, D., Bolstad, B., Dettling, M., Dudoit, S., Ellis, B., Gautier, L., Ge, Y., and Gentry, J., "Bioconductor: open software development for computational biology and bioinformatics," *Genome Biology*, vol. 5, no. R80, 2004.

[48] Geraci, F., Pellegrini, M., and Renda, M., "AMIC@: All Microarray Clusterings @ once," *Nucleic Acids Research*, vol. 36, no. Web Server, pp. W315–W319, 2008.

[49] Goksel, G., Taneli, F., Uslu, R., Ulman, C., Dinc, G., Coskun, T., and Kandiloglu, A., "Serum her-2/neu and survivin levels and their relationship to histological parameters in early-stage breast cancer," *J Int Med Res*, vol. 35, no. 2, pp. 165–172, 2007.

[50] Golub, T., Slonim, D., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J., Coller, H., Loh, M., Downing, J., Caligiuri, M., Bloomfield, C., and Lander, E., "Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring," *Science*, vol. 286, pp. 531–537, 1999.

[51] Hamm, A., Veeck, J., Bektas, N., Wild, P., Hartmann, A., Heindrichs, U., Kristiansen, G., Werbowetski-Ogilvie, T., Del Maestro, R., Knuechel, R., and Dahl, E., "Frequent expression loss of Inter-alpha-trypsin inhibitor heavy chain (ITIH) genes in multiple human solid tumors: A systematic expression analysis," *BMC Cancer*, vol. 8, no. 25, 2008.

[52] Hastie, T., Tibshirani, R., and Friedman, J., *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2001.

[53] Hess, K., Anderson, K., Symmans, W., Valero, V., Ibrahim, N., Mejia, J., Booser, D., Theriault, R., Buzdar, A., Dempsey, P., Rouzier, R., Sneige, N., Ross, J., Vidaurre, T., Gomez, H., Hortobagyi, G., and Pusztai, L., "Pharmacogenomic Predictor of Sensitivity to Preoperative Chemotherapy With Paclitaxel and Fluorouracil, Doxorubicin, and Cyclophosphamide in Breast Cancer," *J Clin Oncol*, vol. 24, no. 26, pp. 4236–4244, 2006.

[54] Hill, A., Brown, E., Whitley, M., Tucker-Kellogg, G., Hunter, C., and Slonim, D., "Evaluation of normalization procedures for oligonucleotide array data based on spiked cRNA controls," *Genome Biology*, vol. 2, no. 12, 2001.

[55] Hofer, M., Kuefer, R., Varambally, S., Li, H., Ma, J., Shapiro, G., Gschwend, J., Hautmann, R., Sanda, M., Giehl, K., Menke, A., Chinnaiyan, A., and Rubin, M., "The Role of Metastasis-Associated Protein 1 in Prostate Cancer Progression," *Cancer Research*, vol. 64, no. 3, pp. 825–829, 2004.

[56] Hoffmann, R., Seidl, T., and Dugas, M., "Profound effect of normalization on detection of differentially expressed genes in oligonucleotide microarray data analysis," *Genome Biology*, vol. 3, no. 7, 2002.

[57] Hua, J., Xiong, Z., Lowey, J., Suh, E., and Dougherty, E., "Optimal number of features as a function of sample size for various classification rules," *Bioinformatics*, vol. 21, no. 8, pp. 1509–1515, 2005.

[58] Hubbell, E., Liu, W.-M., and Mei, R., "Robust estimators for expression analysis," *Bioinformatics*, vol. 18, no. 12, pp. 1585–1592, 2002.

[59] Hull, D., Wolstencroft, K., Stevens, R., Goble, C., Pocock, M., Li, P., and Oinn, T., "Taverna: a tool for building and running workflows of services," *Nucleic Acids Research*, vol. 34, no. Web Server, pp. W729–W732, 2006.

[60] Huttenhower, C., Hibbs, M., Myers, C., and Troyanskaya, O., "A scalable method for integration and functional analysis of multiple microarray datasets," *Bioinformatics*, vol. 22, no. 23, pp. 2890–2897, 2006.

[61] Inza, I., Larranaga, P., Blanco, R., and Cerrolaza, A., "Filter versus wrapper gene selection approaches in DNA microarray domains," *Artificial Intelligence in Medicine*, vol. 31, pp. 91–103, 2004.

[62] Irizarry, R., Bolstad, B., Collin, F., Cope, L., Hobbs, B., and Speed, T., "Summaries of Affymetrix GeneChip probe level data," *Nucleic Acids Research*, vol. 31, no. 4, 2003.

[63] Irizarry, R., Hobbs, B., Collin, F., Beazer-Barclay, Y., Antonellis, K., Scherf, U., and Speed, T., "Exploration, normalization, and summaries of high density oligonucleotide array probe level data," *Biostatistics*, vol. 4, no. 2, pp. 249–264, 2003.

[64] Irizarry, R., Warren, D., Spencer, F., Kim, I., Biswal, S., Frank, B., Gabrielson, E., Garcia, J., Geoghegan, J., Germino, G., Griffin, C., Hilmer, S., Hoffman, E., Jedlicka, A., Kawasaki, E., Martnez-Murillo, F., Morsberger, L., Lee, H., Petersen, D., Quackenbush, J., Scott, A., Wilson, M., Yang, Y., Ye, S., and Yu, W., "Multiple-laboratory comparison of microarray platforms," *Nature Methods*, vol. 2, no. 5, pp. 345–349, 2005.

[65] Ivanov, S., Ivanova, A., Salnikow, K., Timofeeva, O., Subramaniam, M., and Lerman, M., "Two novel VHL targets, TGFBI (BIGH3) and its transactivator KLF10, are up-regulated in renal clear cell carcinoma and other tumors," *Biochem Biophys Res Commun*, vol. 370, no. 4, pp. 536–540, 2008.

[66] Ivliev, A., 't Hoen, P., Villerius, M., den Dunnen, J., and Brandt, B., "Microarray retriever: a web-based tool for searching and large scale retrieval of public microarray data," *Nucleic Acids Research*, vol. 36, no. Web Server, pp. W327–W331, 2008.

[67] Jones, J., Otu, H., Spentzos, D., Kolia, S., Inan, M., Beecken, W., Fellbaum, C., Gu, X., Joseph, M., Pantuck, A., Jonas, D., and Libermann, T., "Gene signatures of progression and metastatis in renal cell cancer," *Clin Cancer Res*, vol. 11, pp. 5730–5739, August 2005.

[68] Jones, W., Wang, M., Moffitt, R., Phan, J., Stokes, T., Bao, W., Wolfinger, R., Li, L., and Parker, J., "The impact of quality-related artifacts on the predictive performance of Affymetrix GeneChips: a robustness case study within MAQC-II," *Submitted to Nat Biotechnol*, 2009.

[69] Keller, A., Backes, C., Al-Awadhi, M., Gerasch, A., Kuntzer, J., Kohlbacher, O., Kaufmann, M., and Lenhof, H.-P., "GeneTrailExpress: a web-based pipeline for the statistical evaluation of microarray experiments," *BMC Bioinformatics*, vol. 9, no. 552, 2008.

[70] Kluger, Y., Yu, H., Qian, J., and Gerstein, M., "Relationship between gene co-expression and probe localization on microarray slides," *BMC Genomics*, vol. 4, no. 49, 2003.

[71] Kong, S., Pu, W., and Park, P., "A multivariate approach for integrating genome-wide expression data and biological knowledge," *Bioinformatics*, vol. 22, no. 19, pp. 2373–2380, 2006.

[72] Kuffner, R., Fundel, K., and Zimmer, R., "Expert knowledge without the expert: integrated analysis of gene expression and literature to derive active functional contexts," *Bioinformatics*, vol. 21, pp. 259–267, 2005.

[73] Kun, L., Ray, P., Merrell, R., and Kwankam, S., "Improving the Health Care and Public Health Critical Infrastructure," *IEEE Engineering in Medicine and Biology Magazine*, vol. 27, no. 6, pp. 21–25, 2008.

[74] Kunz, I., Lin, M., and Frey, L., "Metadata mapping and reuse in cabig," *BMC Bioinformatics*, vol. 10, no. Suppl 2, 2009.

[75] Laiho, P., Kokko, A., Vanharanta, S., Salovaara, R., Sammalkorpi, H., Jarvinen, H., Mecklin, J., Karttunen, T., Tuppurainen, K., Davalos, V., Schwartz Jr, S., Arango, D., Makinen, M., and Aaltonen, L., "Serrated carcinomas form a subclass of colorectal cancer with distinct molecular basis," *Oncogene*, vol. 26, no. 2, pp. 312–320, 2007.

[76] Lasko, T., Bhagwat, J., Zou, K., and Ohno-Machado, L., "The use of receiver operating characteristic curves in biomedical informatics," *Journal of Biomedical Informatics*, vol. 38, no. 5, pp. 404–415, 2005.

[77] Lee, J., Lee, J., Park, M., and Song, S., "An extensive comparison of recent classification tools applied to microarray data," *Computational Statistics and Data Analysis*, vol. 48, no. 4, pp. 869–885, 2005.

[78] Li, C., "Automating dChip: towards reproducible sharing of microarray data analysis," *BMC Bioinformatics*, vol. 9, no. 231, 2008.

[79] Li, C. and Wong, W., "Model-based analysis of oligonucleotide arrays: expression index computation and outlier detection," *PNAS*, vol. 98, pp. 31–36, 2001.

[80] Linehan, W. and Zbar, B., "Focus on kidney cancer," *Cancer Cell*, vol. 6, no. 3, pp. 223–228, 2004.

[81] Liu, R., Wang, X., Chen, G., Dalerba, A., Gurney, A., Hoey, T., Sherlock, G., Lewicki, J., Shedden, K., and Clarke, M., "The prognostic role of a gene signature from tumorigenic breast-cancer cells," *N Engl J Med*, vol. 356, pp. 217–226, 2007.

[82] Loo, L.-H., Roberts, S., Hrebien, L., and Kam, M., "New criteria for selecting differentially expressed genes: Filter-based feature selection techniques for better detection of changes in the distributions of expression levels," *IEEE Engineering in Medicine and Biology Magazine*, vol. 26, no. 2, pp. 17–26, 2007.

[83] Lu, Y., Cai, Z., Xiao, G., Liu, Y., Keller, E., Yao, Z., and Zhang, J., "CCR2 Expression Correlates With Prostate Cancer Progression," *J Cell Biochem*, vol. 101, no. 3, pp. 676–685, 2007.

[84] Lu, Y., Liu, P., Xiao, P., and Deng, H., "Hotelling's T^2 multivariate profiling for detecting differential expression in microarrays," *Bioinformatics*, vol. 21, pp. 3105–3113, 2005.

[85] Luo, J., Zha, S., Gage, W., Dunn, T., Hicks, J., Bennett, C., Ewing, C., Platz, E., Ferdinandusse, S., Wanders, R., Trent, J., Isaacs, W., and De Marzo, A., "Alpha-methylacyl-CoA racemase: a new molecular marker for prostate cancer," *Cancer Research*, vol. 62, no. 8, pp. 2220–2226, 2002.

[86] Maere, S., Heymans, K., and Kuiper, M., "BiNGO: a Cytoscape plugin to assess overrepresentation of Gene Ontology categories in Biological Networks," *Bioinformatics*, vol. 21, no. 16, pp. 3448–3449, 2005.

[87] McConnell, P., Dash, R., Chilukuri, R., Pietrobon, R., Johnson, K., Annechiarico, R., and Cuticchia, A., "The cancer translational research informatics platform," *BMC Med Inform Decis Mak*, vol. 8, no. 60, 2008.

[88] Michiels, S., Koscielny, S., and Hill, C., "Prediction of cancer outcome with microarrays: a multiple random validation strategy," *Lancet*, vol. 365, pp. 488–492, 2005.

[89] Moller-Levet, C., West, C., and Miller, C., "Exploiting sample variability to enhance multivariate analysis of microarray data," *Bioinformatics*, vol. 23, no. 20, pp. 2733–2744, 2007.

[90] Morey, J., Ryna, J., and Van Dolah, F., "Microarray validation: factors influencing correlation between oligonucleotide microarrays and real-time PCR," *Biol Proced Online*, vol. 8, no. 1, pp. 175–193, 2006.

[91] Mosquera, J. and Sanchez-Pla, A., "SerbGO: searching for the best GO tool," *Nucleic Acids Research*, vol. 36, no. Web Server, pp. W368–W371, 2008.

[92] Mramor, M., Leban, G., Demsar, J., and Zupan, B., "Conquering the Curse of Dimensionality in Gene Expression Cancer Diagnosis: Tough Problem, Simple Models," *LNCS*, vol. 3581, pp. 514–523, 2005.

[93] Mukherjee, S. and Roberts, S., "A theoretical analysis of the selection of differentially expressed genes," *J. Bioinformatics Comput. Biol.*, vol. 3, pp. 627–643, June 2005.

[94] Myers, C., Dunham, M., Kung, S., and Troyanskaya, O., "Accurate detection of aneuploidies in array cgh and gene expression microarray data," *Bioinformatics*, vol. 20, no. 18, pp. 3533–3543, 2004.

[95] National Cancer Institute, "Cancer Biomedical Informatics Grid (caBIG)." https://cabig.nci.nih.gov/, 2009.

[96] Nofech-Mozes, S., Trudeau, M., Kahn, H., Dent, R., Rawlinson, E., Sun, P., Narod, S., and Hanna, W., "Patterns of recurrence in the basal and non-basal subtypes of triple-negative breast cancers," *Breast Cancer Res Treat*, 2009.

[97] Ntzani, E. and Ioannidis, J., "Predictive ability of DNA microarrays for cancer outcomes and correlates: an empirical assessment," *Lancet*, vol. 362, pp. 1439–1444, 2003.

[98] Ochs, M. and Casagrande, J., "Information systems for cancer research," *Cancer Invest*, vol. 26, no. 10, pp. 1060–1067, 2008.

[99] Ohmann, C. and Kuchinke, W., "Future developments of medical informatics from the viewpoint of networked clinical research: Interoperability and integration," *Methods Inf Med*, vol. 48, no. 1, pp. 45–54, 2009.

[100] Park, T., Yi, S.-G., Shin, Y., and Lee, S., "Combining multiple microarrays in the presense of controlling variables," *Bioinformatics*, vol. 22, pp. 1682–1689, 2006.

[101] Parkinson, H., Kapushesky, M., Shojatalab, M., Abeygunawardena, N., Coulson, R., Farne, A., Holloway, E., Kolesnykov, N., Lilja, P., Lukk, M., Mani, R., Rayner, T., Sharma, A., William, E., Sarkans, U., and Brazma, A., "ArrayExpress-a public database of microarray experiments and gene expression profiles," *Nucleic Acids Research*, vol. 35, no. Database, pp. D747–D750, 2007.

[102] Parry, R., Jones, W., Stokes, T., Phan, J., Moffitt, R., Fang, H., Shi, L., Oberthuer, A., Fischer, M., Tong, W., and Wang, M., "K-nearest neighbors (KNN) models for microarray gene-expression analysis and reliable clinical outcome prediction," *Submitted to Nat Biotechnol*, 2009.

[103] Patterson, T., Lobenhofer, E., Fulmer-Smentek, S., Collins, P., Chu, T., Bao, W., Fang, H., Kawasaki, E., Hager, J., Tikhonova, I., Walker, S., Zhang, L., Hurban, P., De Longueville, F., Fuscoe, J., Tong, W., Shi, L., and Wolfinger, R., "Performance comparison of one-color and two-color platforms within the MicroArray Quality Control (MAQC) project," *Nat Biotechnol*, vol. 24, pp. 1140–1150, 2006.

[104] Phan, J., Moffitt, R., Barrett, A., and Wang, M., "Improving Microarray Sample Size Using Bootstrap Data Combination," *International Multisymposiums on Computer and Computational Sciences*, pp. 37–44, 2008.

[105] Phan, J., Moffitt, R., Stokes, T., Liu, J., Young, A., Nie, S., and Wang, M., "Biomarkers, Nanotechnology, and Personalized Medicine," *Trends in Biotechnology, In Press*, 2009.

[106] Phan, J., Yin-Goen, Q., Young, A., and Wang, M., "Improving the Efficiency of Biomarker Identification Using Biological Knowledge," *Pac Symp Biocomput*, vol. 14, pp. 427–438, 2009.

[107] Phan, J., Young, A., and Wang, M., "Selecting Clinically-Driven Biomarkers for Cancer Nanotechnology," *Engineering in Medicine and Biology Society*, pp. 3317–3320, 2006.

[108] Pirooznia, M., Gong, P., Yang, J., Yang, M., Perkins, E., and Deng, Y., "ILOOP - a web application for two-channel microarray interwoven loop design," *BMC Genomics*, vol. 9, no. S11, 2008.

[109] Polo, J., Juszczynski, P., Monti, S., Cerchietti, L., Ye, K., Greally, J., Shipp, M., and Melnick, A., "Transcriptional signature with differential expression of BCL6 target genes accurately identifies BCL6-dependent diffuse large B cell lymphomas," *PNAS*, vol. 104, pp. 3207–3212, 2007.

[110] Prowatke, I., Devens, F., Benner, A., Grone, E., Mertens, D., Grone, H.-J., Lichter, P., and Joos, S., "Expression analysis of imbalanced genes in prostate carcinoma using tissue microarrays," *Br J Cancer*, vol. 96, no. 1, pp. 82–88, 2007.

[111] Quackenbush, J., "Microarray Analysis and Tumor Classification," *NEJM*, vol. 354, no. 23, pp. 2463–2472, 2006.

[112] Rai, A., "Biomarkers in translational research: focus on discovery, development and translation of protein biomarkers to clinical immunoassays," *Expert Rev Mol Diagn*, vol. 7, no. 5, pp. 545–553, 2007.

[113] Rainer, J., Sanchez-Cabo, F., Stocker, G., Sturn, A., and Trajanoski, Z., "CARMAweb: comprehensive R- and bioconductor-based web service for microarray data analysis," *Nucleic Acids Research*, vol. 34, no. Web Server, pp. W498–W503, 2006.

[114] Rehrauer, H., Zoller, S., and Schlapbach, R., "MAGMA: analysis of two-channel microarrays made easy," *Nucleic Acids Research*, vol. 35, no. Web Server, pp. W86–W90, 2007.

[115] Reich, M., Liefeld, T., Gould, J., Lerner, J., Tamayo, P., and Mesirov, J., "GenePattern 2.0," *Nature Genetics*, vol. 38, no. 5, pp. 500–501, 2006.

[116] Rohan, S., Tu, J., Kao, J., Mukherjee, P., Campagne, F., Zhou, X., Hyjek, E., Alonso, M., and Chen, Y., "Gene expression profiling separates chromophobe renal cell carcinoma from oncocytoma and identifies vesicular transport and cell junction proteins as differentially expressed genes," *Clin Cancer Res*, vol. 12, pp. 6937–6945, December 2006.

[117] Rosendahl, A. and Forseberg, G., "IGF-I and IGFBP-3 augment transforming growth factor-beta actions in human renal carcinoma cells," *Kidney International*, vol. 70, no. 9, pp. 1584–1590, 2006.

[118] Rosenfeld, N., Aharonov, R., Meiri, E., Rosenwald, S., Spector, Y., Zepeniuk, M., Benjamin, H., Shabes, N., Tabak, S., Levy, A., Lebanony, D., Goren, Y., Silberschein, E., Targan, N., Ben-Ari, A., Gilad, S., Sion-Vardy, N., Tobar, A., Feinmesser, M., Kharenko, O., Nativ, O., Nass, D., Perelman, M., Yosepovich, A., Shalmon, B., Polak-Charcon, S., Fridman, E., Avniel, A., Bentwich, I., Bentwich, Z., Cohen, D., Chajut, A., and Barshack, I., "MicroRNAs accurately identify cancer tissue origin," *Nat Biotechnol*, vol. 26, no. 4, pp. 462–469, 2008.

[119] Rosenzweig, C., Zhang, Z., Sun, X., Sokoll, L., Osborne, K., Partin, A., and Chan, D., "Predicting Prostate Cancer Biochemical Recurrence Using a Panel of Serum Proteomic Biomarkers," *J Urol*, vol. 181, no. 3, pp. 1407–1414, 2009.

[120] Rouzier, R., Rajan, R., Wagner, P., Hess, K., Gold, D., Stec, J., Ayers, M., Ross, J., Zhang, P., Buchholz, T., Kuerer, H., Green, M., Arun, B., Hortobagyi, G., Symmans, W., and Pusztai, L., "Microtubule-associated protein tau: a marker of paclitaxel sensitivity in breast cancer," *Proc Natl Acad Sci*, vol. 102, no. 23, pp. 8315–8320, 2005.

[121] Sardana, G., Dowell, B., and Diamandis, E., "Emerging biomarkers for the diagnosis and prognosis of prostate cancer," *Clin Chem*, vol. 54, no. 12, pp. 1951–1960, 2008.

[122] Schlomm, T., Hellwinkel, O., Buness, A., Ruschhaupt, M., Lubke, A., Chun, F., Simon, R., Budaus, L., Erbersdobler, A., Graefen, M., Huland, H., Poustka, A., and Sultmann, H., "Molecular Cancer Phenotype in Normal Prostate Tissue," *Eur Urol*, vol. 55, no. 4, pp. 885–891, 2009.

[123] Schrader, A., Lechner, O., Templin, M., Dittmar, K., Machtens, S., Mengel, M., Probst-Kepper, M., Franzke, A., Wollensak, T., Gatzlaff, P., Atzpodien, J., Buer, J., and Lauber, J., "CXCR4/CXCL12 expression and signalling in kidney cancer," *Br J Cancer*, vol. 86, no. 8, pp. 1250–1256, 2002.

[124] Schuetz, A., Yin-Goen, Q., Amin, M., Moreno, C., Cohen, C., Hornsby, C., Yang, W., Petros, J., Issa, M., Pattaras, J., Ogan, K., Marshall, F., and Young, A., "Molecular classification of renal tumors by gene expression profiling," *J Mol Diagn*, 2005.

[125] Seo, J. and Hoffman, E., "Probe set algorithms: is there a rational best bet?," *BMC Bioinformatics*, vol. 7, no. 395, 2006.

[126] Shi, L., Reid, L. H., Jones, W. D., Shippy, R., Warrington, J. A., Baker, S. C., Collins, P. J., de Longueville, F., Kawasaki, E. S., Lee, K. Y., Luo, Y., Sun, Y. A., Willey, J. C., Setterquist, R. A., Fischer, G. M., Tong, W., Dragan, Y. P., Dix, D. J., Frueh, F. W., Goodsaid, F. M., Herman, D., Jensen, R. V., Johnson, C. D., Lobenhofer, E. K., Puri, R. K., Schrf, U., Thierry-Mieg, J., Wang, C., Wilson, M., Wolber, P. K., Zhang, L., Amur, S., Bao, W., Barbacioru, C. C., Lucas, A. B., Bertholet, V., Boysen, C., Bromley, B., Brown, D., Brunner, A., Canales, R., Cao, X. M., Cebula, T. A., Chen, J. J., Cheng, J., Chu, T. M., Chudin, E., Corson, J., Corton, J. C., Croner, L. J., Davies, C., Davison, T. S., Delenstarr, G., Deng, X., Dorris, D., Eklund, A. C., Fan, X. H., Fang, H., Fulmer-Smentek, S., Fuscoe, J. C., Gallagher, K., Ge, W., Guo, L., Guo, X., Hager, J., Haje, P. K., Han, J., Han, T., Harbottle, H. C., Harris, S. C., Hatchwell, E., Hauser, C. A., Hester, S., Hong, H., Hurban, P., Jackson, S. A., Ji, H., Knight, C. R., Kuo, W. P., LeClerc, J. E., Levy, S., Li, Q. Z., Liu, C., Liu, Y., Lombardi, M. J., Ma, Y., Magnuson, S. R., Maqsodi, B., McDaniel, T., Mei, N., Myklebost, O., Ning, B., Novoradovskaya, N., Orr, M. S., Osborn, T. W., Papallo, A., Patterson, T. A., Perkins, R. G., Peters, E. H., Peterson, R., Philips, K. L., Pine, P. S., Pusztai, L., Qian, F., Ren, H., Rosen, M., Rosenzweig, B. A., Samaha, R. R., Schena, M., Schroth, G. P., Shchegrova, S., Smith, D. D., Staedtler, F., Su, Z., Sun, H., Szallasi, Z., Tezak, Z., Thierry-Mieg, D., Thompson, K. L., Tikhonova, I., Turpaz, Y., Vallanat, B., Van, C., Walker, S. J., Wang, S. J., Wang, Y., Wolfinger, R., Wong, A., Wu, J., Xiao, C., Xie, Q., Xu, J., Yang, W., Zhang, L., Zhong, S., Zong, Y., and Slikker, W., J., "The Microarray Quality Control (MAQC) project show inter- and intraplatform reproducibility of gene expression measurements," *Nat Biotechnol*, vol. 24, pp. 1151–1161, September 2006.

[127] Shi, L. and The MicroArray Quality Control (MAQC) Consortium, "The MAQC-II Project: A comprehensive study of common practices for the development and validation of microarray-based predictive models," *Submitted to Nat Biotechnol*, 2009.

[128] Sima, C., Braga-Neto, U., and Dougherty, E., "Superior feature-set ranking for small samples using bolstered error estimation," *Bioinformatics*, vol. 21, pp. 1046–1054, 2005.

[129] Simon, I., Katsaros, D., Rigault de la Longrais, I., Massobrio, M., Scorilas, A., Kim, N., Sarno, M., Wolfert, R., and Diamandis, E., "B7-H4 is over expressed in early-stage ovarian cancer and is independent of CA 125 expression," *Gynecol Oncol*, 2007.

[130] Simon, I., Liu, Y., Krall, K., Urban, N., Wolfert, R., Kim, N., and McIntosh, M., "Evaluation of the novel serum markers B7-H4, Spondin 2, and DcR3 for diagnosis and early detection of ovarian cancer," *Gynecol Oncol*, vol. 106, no. 1, pp. 112–118, 2007.

[131] Simon, R., "Diagnostic and prognostic prediction using gene expression profiles in high-dimensional microarray data," *Br J Cancer*, vol. 89, pp. 1599–1604, 2003.

[132] Simon, R., Radmacher, M., Dobbin, K., and McShane, L., "Pitfalls in the Use of DNA Microarray Data for Diagnostic and Prognostic Classification," *Journal of the National Cancer Institute*, vol. 95, no. 1, pp. 14–18, 2003.

[133] Singh, D., Febbo, P., Ross, K., Jackson, D., Manola, J., Ladd, C., Tamayo, P., Renshaw, A., D'Amico, A., Richie, J., Lander, E., Loda, M., Kantoff, P., Golub, T., and Sellers, W., "Gene expression correlates of clinical prostate cancer behavior," *Cancer Cell*, vol. 1, pp. 203–209, 2002.

[134] Somorjai, R., Dolenko, B., and Baumgartner, R., "Class prediction and discovery using gene microarray and proteomics mass spectroscopy data: curses, caveats, cautions," *Bioinformatics*, vol. 19, no. 12, pp. 1484–1491, 2003.

[135] Song, J., Maghsoudi, K., Li, W., Fox, E., Quackenbush, J., and Liu, X., "Microarray blob-defect removal improves array analysis," *Bioinformatics*, vol. 23, no. 8, pp. 966–971, 2007.

[136] Sreekumar, A., Poisson, L., Rajendiran, T., Khan, A., Cao, Q., Yu, J., Laxman, B., Mehra, R. Lonigro, R., Li, Y., Nyati, M., Ahsan, A., Kalyana-Sundaram, S., Han, B., Cao, X., Byun, J., Omenn, G., Ghosh, D., Pennathur, S., Alexander, D., Berger, A., Shuster, J., Wei, J., Varambally, S., Beecher, C., and Chinnaiyan, A., "Metabolomic profiles delineate potential role for sarcosine in prostate cancer progression," *Nature*, vol. 457, no. 7231, pp. 910–914, 2009.

[137] Stokes, T., Moffitt, R., Phan, J., and Wang, M., "chip artifact CORRECTion (caCORRECT): A Bioinformatics System for Quality Assurance of Genomics and Proteomics Array Data," *Annals of Biomedical Engineering*, vol. 35, pp. 1068–1080, 2007.

[138] Stokes, T., Torrance, J., Li, H., and Wang, M., "ArrayWiki: an enabling technology for sharing public microarray data repositories and meta-analysis," *BMC Bioinformatics*, vol. 9, no. S18, 2008.

[139] Tarraga, J., Medina, I., Carbonell, J., Huerta-Cepas, J., Minguez, P., Alloza, E., Al-Shahrour, F., Vegas-Azcarate, S., Goetz, S., Escobar, P., Garcia-Garcia, F., Conesa, A., Montaner, D., and

Dopazo, J., "GEPAS, a web-based tool for microarray data analysis and interpretation," *Nucleic Acids Research*, vol. 36, no. Web Server, pp. W308–W314, 2008.

[140] Troyanskaya, O., Garber, M., Brown, P., Botstein, D., and Altman, R., "Nonparametric methods for identifying differentially expressed genes in microarray data," *Bioinformatics*, vol. 18, pp. 1454–1461, May 2002.

[141] Tusher, V., Tibshirani, R., and Chu, G., "Significance analysis of microarrays applied to the ionizing radiation response," *PNAS*, vol. 98, pp. 5116–5121, 2001.

[142] VanGuilder, H., Vrana, K., and Freeman, W., "Twenty-five years of quantitative PCR for gene expression analysis," *BioTechniques*, vol. 44, no. 5, pp. 619–616, 2008.

[143] van't Veer, L., Dai, H., van de Vijver, M., He, Y., Hart, A., Mao, M., Peterse, H., van der Kooy, K., Marton, M., Witteveen, A., Schreiber, G., Kerkhoven, R., Roberts, C., Linsley, P., Bernards, R., and Friend, S., "Gene expression profiling predicts clinical outcome of breast cancer," *Nature*, vol. 415, pp. 530–536, 2002.

[144] Varambally, S., Yu, J., Laxman, B., Rhodes, D., Mehra, R., Tomlins, S., Shah, R., Chandran, U., Monzon, F., Becich, M., Wei, J., Pienta, K., Ghosh, D., Rubin, M., and Chinnaiyan, A., "Integrative genomic and proteomic analysis of prostate cancer reveals signatures of metastatic progression," *Cancer Cell*, vol. 8, pp. 393–406, 2005.

[145] Varma, S. and Simon, R., "Bias in error estimation when using cross-validation for model selection," *BMC Bioinformatics*, vol. 7, no. 91, 2006.

[146] Vo, T., Phan, J., Kiet, N., and Wang, M., "Reproducibility of Differential Gene Detection across Multiple Microarray Studies," *Engineering in Medicine and Biology Society*, pp. 4231–4234, 2007.

[147] Wang, J., Coombes, K., Highsmith, W., Keating, M., and Abruzzo, L., "Differences in gene expression between B-cell chronic lymphocytic leukemia and normal B cells: a meta-analysis of three microarray studies," *Bioinformatics*, vol. 20, pp. 3166–3178, 2004.

[148] West, M., Blanchette, C., Dressman, H., Huang, E., Ishida, S., Spang, R., Zuzan, H., Olson, J., Marks, J., and Nevins, J., "Predicting the clinical status of human breast cancer by using gene expression profiles," *PNAS*, vol. 98, no. 20, pp. 11462–11467, 2001.

[149] Whitehall, B. and Lu, S., *Machine Learning: A Multistrategy Approach*. 1994.

[150] Xiao, Y., Frisina, R., Gordon, A., Klebanov, L., and Yakovlev, A., "Multivariate search for differentially expressed gene combinations," *BMC Bioinformatics*, vol. 5, no. 164, 2004.

[151] Xiong, M., Fang, X., and Zhao, J., "Biomarker Identification by Feature Wrappers," *Genome Research*, vol. 11, pp. 1878–1887, 2001.

[152] Xu, L., Tan, A., Naiman, D., Geman, D., and Winslow, R., "Robust prostate cancer marker genes emerge from direct integration of inter-study microarray data," *Bioinformatics*, vol. 21, pp. 3905–3911, 2005.

[153] Yoon, S., Yang, Y., Choi, J., and Seong, J., "Large scale data mining approach for gene-specific standardization of microarray gene expression data," *Bioinformatics*, vol. 22, no. 23, pp. 2898–2904, 2006.

[154] Zeeberg, B., Feng, W., Wang, G., Wang, M., Fojo, A., Sunshine, M., Narasimhan, S., Kane, D., Reinhold, W., Lababidi, S., Bussey, K., Riss, J., Barrett, J., and Weinstein, J., "GoMiner: A Resource for Biological Interpretation of Genomic and Proteomic Data," *Genome Biology*, vol. 4, no. R28, 2003.

[155] Zeeberg, B., Qin, H., Narasimhan, S., Sunshine, M., Cao, H., Kane, D., Reimers, M., Stephens, R., Bryant, D., Burt, S., Elnekave, E., Hari, D., Wynn, T., Cunningham-Rundles, C., Stewart, D., Nelson, D., and Weinstein, J., "High-Throughput GoMiner, an 'industrial-strength' integrative Gene Ontology tool for interpretation of multiple-microarray experiments, with application to studies of Common Variable Immune Deficiency (CVID)," *BMC Bioinformatics*, vol. 6, no. 168, 2005.

[156] Zhang, X., "Biomarker validation: movement towards personalized medicine," *Expert Rev Mol Diagn*, vol. 7, no. 5, pp. 469–471, 2007.

[157] Zheng, Q. and Wang, X., "GOEAST: a web-based software toolkit for Gene Ontology enrichment analysis," *Nucleic Acids Research*, vol. 36, no. Web Server, pp. W358–W363, 2008.

[158] Zhu, Y., Davis, S., Stephens, R., Meltzer, P., and Chen.Y., "GEOmetadb: powerful alternative search engine for the Gene Expression Omnibus," *Bioinformatics*, vol. 24, no. 23, pp. 2798–2800, 2008.

VITA

John H. Phan was born in Oklahoma City, OK and attended the University of Oklahoma as a Regents Academic Scholar. He graduated *Summa Cum Laude* with a Computer Engineering degree in 2003. As an undergraduate, John focused on software engineering and computer architecture, but began to shift his attention to biomedical related research. At the University of Oklahoma Health Sciences Center, John worked as an undergraduate researcher for several semesters, focusing on data management and processing of medical images. This ultimately led him to the Biomedical Engineering department at the Georgia Institute of Technology and Emory University in the fall of 2003. In the fall of 2004, John officially joined the Bio-Medical Informatics and Bio-imaging Laboratory (Bio-MIBLab) under the advisement of Dr. May D. Wang. He immediately began his research in the areas of biomarker identification, machine learning, and high-performance computing. This work eventually shifted to knowledge-guided machine learning and translational bioinformatics, the primary focus of John's PhD dissertation, which he completed in the spring of 2009.